Report No. WD-9758-159-2
June 23, 1969

A GENERALIZED ATTITUDE DETERMINATION SYSTEM

Prepared For
Clyde H. Freeman
Code 565
Goddard Space Flight Center
Greenbelt, Maryland 20771

PREPARED BY: John P. Crocker
John P. Crocker, CSC

REVIEWED BY: R E Trettau
Ronald E. Trettau, CSC

APPROVED BY: Clyde H. Freeman
Clyde H. Freeman, GSFC

COMPUTER SCIENCES CORPORATION
8121 Georgia Avenue
Silver Spring, Maryland 20910

TABLE OF CONTENTS

PART 1 - GENERAL DISCUSSION

TABLE OF CONTENTS (Cont'd)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Cont'd)

SECTION 1

INTRODUCTION

The Generalized Attitude Determination System (GADS) is designed for use in attitude determination problems where there is a distinct need for differential correction; that is, where there is a need to "refine" a set of parameters. This program solves the problem of determining the orientation of a spacecraft, given observed data which may be obtained from the ground and/or the on-board instruments. Note, however, that the GADS system is a general-purpose, least-squares, differential correction program, and proceeds according to Donald Marquardt's technique, Reference 2, which is discussed in Section 5. As illustrated in Figure 4, only the low-level worker modules qualify it as an attitude program.

This program is written primarily in FORTRAN V with few special-purpose modules written in SLEUTH II. The resident executive is the UNIVAC EXEC VIII. The hardware requirements are a UNIVAC 1108 with FASTRAND. Core requirements are approximately 25,000 and 31,000 computer instructions and data cells, respectively (decimal).

GADS is designed to function as a subroutine and is controlled with simple English-like statements. The input to the program must include observed data provided by the user through a COMMON area. Thus, GADS assumes no responsibility for I/O action concerning the observations. The layout rules for data are considered below.

In addition to observational data, most problems require additional independent inputs such as environmental data. Hence, provision has been made for reading and interpolating orbit data from FASTRAND. Environmental variables may be computed internally so that I/O action is unnecessary. GADS can, for example, generate its own solar line-of-sight vector (module GADSSE).

GADS is also designed for adaptability; that is, for ease in handling new and unexpected problems. While most requirements will be met by the available capabilities, provision has been made for adding new modules and for expanding the scope of the aforementioned English-like controls. Some requirements for future programming have been anticipated and are discussed where pertinent. Refer especially to Section 6. Instructions for future programming and for expanding the scope of these controls are also given in Section 3. It will be seen that adding a new option can sometimes be done by analogy with the existing options. When future capabilities are described, they are identified by the comment "not ready" or "n.r."

SECTION 2

USAGE

2.1 INPUTS

2.1.1 OBSERVATIONAL INPUTS

Upon entry the GADS system expects to find the observations in the array COMMON/COMM1/COMM(4000), as illustrated in Figure 5. Data points from any one sensor are placed in an array, preferably without skipping computer cells. (When a cell is to be ignored, it must contain a -0.0 and should be counted as an observation.)

The relative location (with respect to the first word of COMM) of the first word of each raw data array is provided by the user in the array LOCATE. Similarly, the number of data points in each string of observed data is provided in the array LENGTH. Thus, the GADS system can access these raw data by means of the numbers contained in LENGTH and LOCATE which are normally determined in the run deck discussed later. The user may, if he wishes, load LENGTH and LOCATE during preprocessing. Dynamic core allocation can, in this way, be achieved within the array COMM.

The user also provides the sample times as shown in Figure 5. The relative address of a sample time is obtained by adding the constant JTIME to the relative address of the corresponding data. Thus the layout of the array of sample times is identical with that of the observational data. The variable JTIME is computed in GADS from additional data furnished by the user, namely LOCA and LOCT. Refer to paragraph 2.6.

2.1.2 ENVIRONMENTAL INPUTS

In most attitude determination problems, environmental data is obtained via magnetic tape. For best results, the contents of the tape should be placed in a FASTRAND file. The GADS system will then read the file as required. This data is placed in array COMMON/ORBIT/ORBIT (I, J) by module GADSAO. Interpolation can then be performed by module GADSSO. The dimensions I and J are determined by the user's tape records. Hence, the user should insure that the ORBIT arrays in these modules are compatible.

2.1.3 IMPLICIT INPUTS

Several variables should be provided implicitly, i.e., through COMMON storage. These variables control printouts, simulation, tape record layouts, and others. This discussion is reserved for the summary of input requirements, paragraph 2.6.

2.1.4 CARD INPUTS

Module GADSIN will always read cards on the first entry. These cards constitute the "run deck" which is fully explained in Section 3. It should be emphasized, however, that practically the entire run deck can be entered by means of data statements. In such a case, GADSIN would merely read two cards, namely:

START GADS CARDS

STOP GADS CARDS.

The term "run deck" refers to the cards themselves or their images in core, as the case may be.

2.2 <u>GADS RUN DECK</u>

A full description of the run deck is given in Section 3. At this point the fundamentals are introduced.

2.2.1 PURPOSE

The purpose of the GADS run deck is to:

- Specify sensor characteristics and configuration
- Specify the method of solution
- Provide convenience and flexibility with respect to changing requirements, program checkout, and method of approach

With some loss of convenience, most of the contents of the run deck may be incorporated into the program.

Sensor characteristics and configuration pertain to physical properties, such as: type of output function, type of sensitivity, and the mounting position of the sensor with respect to the main body frame of reference. These concepts are amplified later to make the sensor-related input data as self-explanatory as possible.

The phrase "method of approach" means: a) type of parameters employed (Euler angles, Euler parameters, roll-pitch, and yaw, etc.), b) type of motion (simple force-free spin, force-free balanced precessional motion, forced motion, etc), c) the scheme for "cascading" the parameters and the sensors (refer to Glossary), d) whether engineering units or telemetry counts are to be employed for a given sensor at a given stage of the computation, e) whether sensor calibration constants or mounting constants are to be corrected, f) whether displays are desired, and g) several other options.

2.2.2 THE RUN DECK AS A PROGRAM

It will be seen that the run deck is itself a program, albeit with rudimentary syntax and few operations. Its language is designed to provide a clear and simple way to apply the methods of differential correction with a minimum of drudgery. Hence, it contains those operations and directives needed to apply a general purpose differential correction loop to vector functions in several variables. In addition, several directives are available for ancillary purposes such as the generation of SC-4020 displays and statistical measures of confidence.

2.3 THE GADS LANGUAGE

The GADS run deck consists of "statements" in the GADS language. These statements contain operators, operands, directives and related data. The directives are 18 hollerith characters with strong mnemonic value and are always punched in columns 1 through 18. A card containing a directive is a GADS control card and carries no additional punched data of significance to the program. A GADS control card is usually followed by several data cards containing the elements of a problem. Elements may be "GADS literals" (ordinary floating and fixed point numbers) or symbols. All symbols are 6 hollerith characters.

Some directives are declarative, others executable. The order of declarative directives affects the mathematical definition of the problem while the order of the executable directives is related to the order of execution.

2.4 EXAMPLE

To illustrate the use of GADS, consider the attitude determination of ISIS-A. A flow diagram is given in Figure 6. As implied in that figure, once the GADS run deck is ready and the user has complied with the raw data input requirements, the following statements result in the required attitude calculations.

```
CALL GADSIN ($)
CALL GADS ($,$,N,A)
```

These statements also illustrate the fact that GADSIN must be called each time that the GADS is overlayed, as in the ISIS-A system. When there is no overlay, GADSIN need only be called once. The run deck is read only on the first call. On subsequent calls, GADSIN rescans the card images which are kept in the "root" array RUNDEK.

2.5 OUTPUT

In most cases, the user wants to generate an output tape containing specific attitude data. His output array, A, appears in the calling sequence above as does the total number of item sets, N. The GADS system will compute attitude at the N specified times and call module GADSAI where the user may store those attitude parameters of interest to him. For an illustration of how this is accomplished, see module GADSAI/ISISA.

The $ returns are used in case of errors and in such cases the output attitude will not have been generated. Upon successful return from GADS, the array A will contain the desired attitude data. Generating the output tape is left to the user.

How the N argument times are determined is discussed in the modules GADS/ISISA, GADSAT, and GADSAI/ISISA and is not crucial to the use of GADS. This and other programming details are best understood by these examples which can be found in the program documentation.

2.6 SUMMARY OF BASIC REQUIREMENTS

The basic inputs and rules for the user who plans an executive data reduction program which will use GADS are as follows.

a. Provide communication with GADS by means of the following COMMON variables:

1.) LOCA = relative address, first raw data
2.) LOCT = relative address, first sample time
3.) COMM = storage for the above raw data
4.) TFIXED = double precision millisecond of year time origin or reference time for all sample times
5.) INRSET = reset flag = 1, initially set by a DATA statement in module GADSRD.
6.) LOCEND = maximum space in COMM, i.e., its dimension

The following are optional:

1.) TORIGN = six floating point words containing time origin data (see paragraph 4.3, GADSLS)
2.) When using orbit tapes, provide any variable used in modules of the type GADSSO and GADSAO which are concerning with the layout of tape records (see example GADSSO/EPED)
3.) When simulating, provide any variable used in GADSSS which is concerned with generating artificial data by means of simulation
4.) When using module SHADOW, provide the necessary inputs

b. Provide for an output array, if any, by means of the calling sequence to GADS and GADSAT. Also add any specially desired output attitude items in module GADSAI.

c. Provide the GADS run deck which will be maintained in hollerith form in COMMON/RUNDEK/RUNDEK (14,150) during processing. Note that the run deck may be loaded into this array by means of data statements thereby obviating the need for physically reading a bulky set of cards. (See module GADSRD/ISISA.) It is important that array RUNDEK not be destroyed during overlays when it is loaded from cards.

d. Provide the MAP program containing the following statements in the main segment:

```
IN GADS. GADSDA
IN GADS. GADSDV
IN GADS. GADSRD
```

These will provide the loading of the necessary constants; GADS being the name of the GADS file.

e. Provide the array ORBIT whenever an orbit tape is used. Load the contents of the tape onto FASTRAND and determine that modules GADSAO and GADSSO have compatible COMMON/ORBIT/ statements. The file name should appear in ORBFN.

2.7 SIMULATION

The practiced user may also wish to take advantage of the simulation capability for efficient checkout and other studies. In this case, he is responsible for the inputs to module GADSSS. Note that this module generates synthetic orbital data (sensor operand data) and I/O will result. This module also makes reference to the outputs from module GADSIN, i.e., module GADSSS must not be called before GADSIN. For an example of the use of GADSSS, see module GADS/EPED.

2.8 DESCRIPTION OF PRINTOUT

2.8.1 DESCRIPTION OF PRINTOUT FROM DIFFERENTIAL CORRECTION

2.8.1.1 Core Usage Initialization Report

The initialization module GADSIN prints the summary of the space used by each complex of sensors. All addresses pertain to array COMM and are FORTRAN addresses. They are shown in both octal and decimal form and are useful in inspecting post-mortem dumps and program coding, respectively. Only the most important addresses are given. A graphic explanation of the layout of array COMM is given in Figure 10.

ILLUSTRATION OF CORE SPACE USAGE REPORT

```
CORE SPACE USED BY COMPLEX MAG2
                    OCTAL        AND DECIMAL STARTING LOCATIONS
          DATA      000000000001          1
          TIME      000000002001       1025
          END       000000027340      12000
          TOP       000000006702       3522
          A MATR    000000004771       2553
          Y MATR    000000004155       2157
          TELEM.    000000000000          0
          FREE      000000020436       8478
          F         000000004001       2049
          RESID.    000000004004       2052
          DF/DU     000000004007       2055
          HF        000000004031       2073
          DHF/DU    000000004031       2073
          YI        000000004031       2073
          WORK      000000004125       2133
          AX=Y      000000004301       2241
          FLAG      000000005461       2865
          FILL      000000000000          0
          D.E. F    000000006061       3121
          D.E. G    000000006061       3121
          D.E. P    000000006061       3121
          D.E. Z    000000006061       3121
          AUG G     000000006061       3121
          D.E. E    000000006061       3121
          D.E. 1    000000000000          0
          DEL U     000000000000          0
          ONE       000000000000          0
          NORMLZ    000000004111       2121
          A INV.    000000000000          0
          Y NORM    000000004045       2085
          A NORM    000000004411       2313
          NVI       000000000014         12
```

The meaning of each address is as follows:

Label	Meaning
DATA	Observational input data.
TIME	The corresponding sample times.
END	The size or dimension of COMM.
TOP	The highest address used in the given complex.
A MATR	The matrix of coefficients (double precision) obtained during the first iteration.
Y MATR	The gradient vector (double precision) obtained during the first iteration.
TELEM.	Not used.
FREE	The excess space in COMM.
F	Not used.
RESID.	Not used.
DF/DU	Not used.
HF	Constraint functions.
DHF/DU	Constraint derivatives.
Y1	Working space for the gradient vector (double precision).
WORK	Not used.
AX=Y	Working space for the matrix of coefficients (double precision).
FLAG	Outlier detection flags.
FILL	Not used.
D. E. F	Differential equations: First integrals.
D. E. G	Differential equations: Derivatives.
D. E. P	Differential equations: Interpolated first integrals.
D. E. Z	Differential equations: Difference tables.
AUX G	Auxiliary functions.
D. E. E	Differential equations: Accuracy parameter.

D. E. 1	Not used.
DEL U	Not used.
ONE	Not used.
NORMLZ	Normalization factors for matrix of coefficients.
A INV.	Not used.
Y NORM	Normalized Y1.
A NORM	Normalized matrix of coefficients.
NV1	Twice the number of active parameters.

2.8.1.2 Differential Correction

The differential correction module GADSLS produces a running commentary as differential correction progresses.

The meaning of each line is as follows:

PROCESS COMPLEX XXXXXX

Meaning: Identifies the given sensor complex by its six-character name.

DAYS HOURS MINUTS MILLS YEARS MS YR

Meaning: This line will contain the time origin of the calculations. The user must supply these numbers if he desires them printed here. See paragraph 2.6.

PARTICIPATING PARAMETERS AND THEIR INITIAL VALUES

Meaning: Introduces the next two lines which are self-evident.

CONVERGENCE REQUIREMENTS

Meaning: The convergence criteria for each of the preceding parameters is printed next.

ITERATION NUMBER N LAMBDA MAX=L

Meaning: This line states the iteration number and the maximum value of LAMBDA. The latter parameter is 2 if the gradient vector is being monitored, and 1 if the vector is not being monitored. See Notes, module GADSLS.

XLAM=XXXX GAMMA=YYYY RSUMSQ=ZZZZ LAMBDA=N

Meaning: This line gives the Marquardt mixture parameter x_λ, the angle γ between the gradient vector and the Taylor vector, the root-mean-squared error (the unbiased standard deviation of fit), and the gradient method-Taylor method pointer LAMBDA, respectively. LAMBDA=1 corresponds to the gradient method; LAMBDA=2, to the Taylor method. When XLAM is very small, LAMBDA max is set to 1 since the comparison between the two methods is not necessary. This will occur when convergence is proceeding efficiently. Refer to the description of the GADSLS module.

DIFFERENTIAL CORRECTIONS ARE

Meaning: This line introduces the next few lines which are obvious. Note that in GADS the corrections are subtracted, not added.

An example of these lines of print are shown below.

2.8.1.3 Convergence in Differential Correction

When differential correction can be terminated successfully, the following lines of print appear:

TOTAL N. DATA PTS. USED N

Meaning: This line states total number of data points included in the given sensor complex.

ILLUSTRATION OF DIFFERENTIAL CORRECTION OPENING STATEMENTS

```
PROCESS COMPLEX MAG2
       1.    DAYS              0.  HOURS           0.  MINUTS          0.  MILLS        1965.  YEARS .00000    MS YR

PARTICIPATING PARAMETERS AND THEIR INITIAL VALUES
    ALPHA         DELTA          PHIO           THETAO        PSIO          PHIDOT
 .1000000+00   .1000000+01   .2000000-00   .1000000+01   .0000000      .1100000+00

CONVERGENCE REQUIREMENTS ARE
 .1110000-03   .1110000-03   .1110000-03   .1110000-03   .1110000-03   .1110000-03

                                                       ITERATION NUMBER  1 LAMBDA MAX= 2

XLAM= .99999999-04 GAMMA= .60628544+02 RSUMSQ= .87932840+02 LAMBDA= 1

DIFFERENTIAL CORRECTIONS ARE
    ALPHA         DELTA          PHIO           THETAO        PSIO          PHIDOT
-.2926682-00   .2825930-00   .9161180-00   .1302398-00   .2203024-01   .9293150-03

XLAM= .99999999-05 GAMMA= .66522050+02 RSUMSQ= .87598839+02 LAMBDA= 2

DIFFERENTIAL CORRECTIONS ARE
    ALPHA         DELTA          PHIO           THETAO        PSIO          PHIDOT
-.4911450-00   .2341303-00   .1152245+01   .1302894-00   .2185945-01   .9277199-03
```

ILLUSTRATION OF PRINTOUT FROM DIFFERENTIAL CORRECTION

```
                                                             ITERATION NUMBER   5 LAMBDA MAX= 2

XLAM= .99999995-03 GAMMA= .58760779+02 RSUMSQ= .10974621+02 LAMBDA= 1

DIFFERENTIAL CORRECTIONS ARE
     ALPHA          DELTA           PHIO           THETAO          PSIO           PHIDOT
-.5833566-01   .2469394-01  -.5585509-01  -.3446827-01   .2735581-00    .9024124-03

XLAM= .99999994-09 GAMMA= .58760779+02 RSUMSQ= .10974621+02 LAMBDA= 2

DIFFERENTIAL CORRECTIONS ARE
     ALPHA          DELTA           PHIO           THETAO          PSIO           PHIDOT
-.5833567-01   .2469394-01  -.5585509-01  -.3446827-01   .2735581-00    .9024124-03

                                                             ITERATION NUMBER   6 LAMBDA MAX= 2

XLAM= .99999994-09 GAMMA= .55465788+02 RSUMSQ= .23704471+01 LAMBDA= 1

DIFFERENTIAL CORRECTIONS ARE
     ALPHA          DELTA           PHIO           THETAO          PSIO           PHIDOT
-.1445487-01   .3638499-02  -.2863383-01  -.4147562-02   .1354538-00    .3920487-03

XLAM= .99999993-10 GAMMA= .55465791+02 RSUMSQ= .23704460+01 LAMBDA= 2

DIFFERENTIAL CORRECTIONS ARE
     ALPHA          DELTA           PHIO           THETAO          PSIO           PHIDOT
-.1445484-01   .3638508-02  -.2863387-01  -.4147562-02   .1354538-00    .3920487-03

                                                             ITERATION NUMBER   7 LAMBDA MAX= 1

XLAM= .99999994-09 GAMMA= .53960776+02 RSUMSQ= .16075005+01 LAMBDA= 1

DIFFERENTIAL CORRECTIONS ARE
     ALPHA          DELTA           PHIO           THETAO          PSIO           PHIDOT
-.1396661-02  -.1016901-02  -.5239181-02   .1085516-02   .2219911-01    .6060317-04

                                                             ITERATION NUMBER   8 LAMBDA MAX= 1

XLAM= .99999994-09 GAMMA= .53197026+02 RSUMSQ= .15998542+01 LAMBDA= 1

DIFFERENTIAL CORRECTIONS ARE
     ALPHA          DELTA           PHIO           THETAO          PSIO           PHIDOT
 .2805539-03  -.3820513-04   .4981026-03   .6993427-04  -.2338162-02  -.6225109-05
```

DEGREES OF FREEDOM N

Meaning: This line states the statistical degrees of freedom. (See Reference 16.)

TOTAL N. OF OUTLIERS

Meaning: The number of data values leading to weighted residuals in excess of X*SDEV, where X is the number entered in card columns 71-80, card G14-1. See Run Deck.
NOTE: The number of outliers will not appear unless outlier detection was requested on the said card.

TOTAL N. OF FILLS

Meaning: This is the total number of fill-data flags encountered. Fill-data flags are discussed in the Glossary and Usage.

STANDARD DEVIATION OF FIT

Meaning: This is the root-mean-squared error (the unbiased standard deviation of the weighted residuals).

The next eight lines give the entire 32 parameters at the conclusion of differential correction. Asterisks printed next to the parameter name indicate that the parameter was active. Refer to illustration on following page.

ILLUSTRATION OF DIFFERENTIAL CORRECTION FINAL RESULTS

```
TOTAL N. DATA PTS. USED      768
DEGREES OF FREEDOM           762
TOTAL N. OF OUTLYERS           0
TOTAL N. OF FILLS              0
STANDARD DEVIATION/FIT     .15997+01
ALPHA   (*)     DELTA   (*)     PHIO    (*)     THETAO (*)      PSIO    (*)     PHIDOT (*)      PSIDOT          AAA
 .1734928-02    .1047752+01    -.3903101-02    .9988903-00     .4101333-02     .1000204+00     .1040000+00     .3740000+01
BBB             CCC             PO              QO              RO              TO              COEFF1          COEFF2
 .4000000+01    .5430000+01     .5430000+01     .3740000+01     .4564000+02     .0000000        .1000000+00     .0000000
COEFF3
-.1000000+01   -.1000000+01    -.1000000+01     .0000000        .0000000        .0000000        .0000000        .0000000

-.1000000+01   -.1000000+01    -.1000000+01     .0000000        .0000000        .0000000        .0000000        .0000000
```

2.8.2 DESCRIPTION OF OUTPUT FROM GADSST

The following description of the output produced by GADSST is illustrated below with an actual computer-produced output derived from a two-parameter fitting process. The mathematical derivation and interpretation of all parameters is contained in Reference 16.

CORRELATION MATRIX CORMAT (I, J)

The parameter names are listed in a column to the left in the order of appearance of their cross-correlations in the matrix. As an example, element (1, 2) (or 2, 1 since matrix is symmetric) in the matrix corresponds to the correlation between parameter 1 and parameter 2. Likewise, for a larger matrix, element (I, J) corresponds to the correlation between parameter I and parameter J, where I and J identify the Ith and Jth parameters in the parameter name column. As discussed in Reference 16, it is desirable to have the off-diagonal terms of the matrix approach zero.

CONFIDENCE LEVEL = .95 T = X.XX

CONFIDENCE LEVEL = .50 T = X.XX

For these two confidence levels, the student "t" statistic is printed. This is the value of T_α (α= 9.5 and .50 in this case) which is used in computing interval estimates of the parameters. Computation of parameter interval estimates is discussed in paragraph 4.3.2 of Reference 16.

PARAMETERS

The parameter names and their final estimated values are listed.

SIGMA P

This is a listing of the unbiased standard deviation (α) of each parameter.

TC

TC is the absolute value of the ratio of the parameter to its standard deviation. It is used to determine when a parameter is significantly different from zero. As an example of its application, if a parameter for bias were included in the fitting process, it may be desirable to check this parameter to determine if it is significantly different from zero.

In the program, all the student "t" values are compared with TC until the largest value less than TC is found, and then its associated confidence level is listed just to the right of TC under the heading LEVEL. Hence, the value under LEVEL indicates the probability level with which the parameter is significantly different from zero.

It should be noted that this application of the t-test of significance applies only to the case where the estimated parameter is to be compared with zero. Reference 16 discusses how similar tests of significance can be performed for other theoretical parameter values.

CONFIDENCE LEVEL = .XX F = X.XX

F is the value of the Snedecor "F" statistic for the associated confidence level. This value of F is used in computing the contour standard deviation.

CONTOUR STANDARD DEVIATION = X.XX

This is the value of the contour standard deviation σ_C as discussed in paragraph 4.3.4 of Reference 16. σ_C is that value of the residual standard deviation which would be expected if the parameters were allowed to vary simultaneously over an ellipsoidal confidence region of the prescribed probability.

P ELLIPSOIDAL ESTIMATES OF PARAMETERS CONTOUR .XX

Listed under P are the final values of the estimated parameters, the names of which appear to the left of these listed values.

Listed under ELLIPSOIDAL ESTIMATES OF PARAMETERS are the vector components which describe the ellipse which has a probability contour as specified by CONTOUR .XX, where .XX is the probability level. This subject of ellipsoidal estimates is discussed in Reference 16, paragraph 4.3.4. Each column under this heading is a vector representing an axis of the error ellipse which has as its center the parameter estimate listed under P. Each element of a particular column is a vector component specifying a distance along an axis, the order being analogous to the order in which the parameters are listed. As an example, the first element of the first column represents a distance along the ALPHA-axis, the second element a distance along the PHI0-axis. The composite of these two components represents one axis of the error-ellipse. Likewise, the second column specifies the second axis of the ellipse. Obviously, there will be as many components of each vector as there are parameters, and also as many vectors as there are parameters.

Example of Printout from GADSST

```
1*****GADSST IN CONTROL ******

CORRELATION MATRIX CORMAT(I,J)
ALPHA          1.0000      .0046
PHIO            .0046     1.0000

CONFIDENCE LEVEL=   95.00  T=       1.97

CONFIDENCE LEVEL=   50.00  T=        .68

PARAMETERS                          SIGMA P          TC      LEVEL
ALPHA          1.59519182-03      9.72045010-04     1.64       50.
PHIO          -1.04968327-03      3.14528520-12**********      95.

CONFIDENCE LEVEL=  .95000  F=     3.09  CONTOUR STANDARD DEVIATION =        2.03971400+00

                                P  ELLIPSOIDAL ESTIMATES OF PARAMETERS CONTOUR  95.
ALPHA              1.59519182-03      1.71216748-03     -1.70424397-03
PHIO              -1.04968327-03      5.54011710-12      5.51450320-12

CONFIDENCE LEVEL=  .50000  F=      .70  CONTOUR STANDARD DEVIATION =        2.02088270+00

                                P  ELLIPSOIDAL ESTIMATES OF PARAMETERS CONTOUR  50.
ALPHA              1.59519182-03      8.13568440-04     -8.09803440-04
PHIO              -1.04968327-03      2.63249030-12      2.62031940-12
```

2.9 OBTAINING SC-4020 HARDCOPY OR MICROFILM

In order to obtain a tape output on Unit H suitable for the SC-4020, simply delete the GADS version of GRID1V and POINTV. This will allow the GSFC modules to be loaded.

2.10 UPDATING THE GADS RUN DECK COMMON BLOCK

GADSRD is a FORTRAN block data subroutine which loads the named common area, RUNDEK, at object time. The contents of this common block are the GADS Run Deck (card images) and various other parameters, all of which must remain undisturbed during the processing. (All other arrays can be overlayed, if desired.)

Since these card images constitute the GADS "program", the question arises as to how a swift modification can be made, pending availability of new FORTRAN data statements. There are four (4) possibilities, they are as follows:

a. The first possibility is that the required modification is so brief that a rapid recompilation is possible. This will only be possible when the change involves simple replacements; i.e., when the total number of card images remains unchanged.

b. The second approach is to ignore the present contents of the module GADSRD and load the entire set of card images at object time using the following control cards:

```
START GADS CARDS
GADS RUN DECK
(GADS Run Deck goes here)
STOP
PRINT RUN DECK
STOP GADS CARDS
```

Though not necessary, the "PRINT" control card is desirable since it provides the user with a comprehensive listing of the contents of the array RUNDEK.

c. The third alternative makes use of an internal updating capability in module GADSIN. The method of updating the card images in array RUNDEK is analogous to the method of updating symbolic programs. That is, one specifies three integers -I, J, K. The meaning of these is as follows:

-I, 0, K:	insert K given card images beginning after the I resident image;
-I, J, K:	delete the I through J resident card images and insert the K given card images;
-I, J, 0:	delete the I through J resident card images with no insertions.

During these operations, the card images are maintained contiguous by shifting up and down as required. The "card excess" or "bias" is also maintained so that the user need not be concerned, as in program updating, with the effect of additions and deletions upon the card count. He is limited, furthermore, to making the modifications in ascending order of card count, also as in program updating.

The format for an UPDATE control card is as follows:

Card Col.	Format	Variable	Comment
1-6	A6	CARD	This field contains "UPDATE".
7-10	4X		Ignored.
11-15	I5	-I	This variable is discussed in the foregoing paragraphs. (This minus sign is optional.)
16-20	I5	J	Also discussed above.
21-25	I5	K	This must be the exact number of card images to be inserted.

A typical set of modification controls is:

```
START GADS CARDS
PRINT RUN DECK
UPDATE  -83  84  2
DISPLAYS
COMPLEX  SUNLIT   FINALS   BYSTEP
STOP UPDATE
PRINT RUN DECK
STOP GADS CARDS
```

This program would replace the 83rd and 84th card images with those provided. In addition, the GADS run deck would be printed before and after the modification. The second printing is good insurance against undetected erroneous updates.

Finally, observe the STOP UPDATE card. This card must follow the last group of updates and is obviously used to halt interpretation of additional updates.

d. In addition, module GADSIN recognizes certain standard GADS control cards during the input stream phase. That is, if any of the following controls appear during the reading of the input card stream and a GADS RUN DECK card has not previously appeared, they will replace the corresponding controls and related specs in array RUNDEK. The groups so recognized are: G3, G4, G5, G6, G7, G8, G9, G17. These groups are discussed below in paragraph 3.4.

A typical use of this approach is to replace the parameter estimates:

```
START GADS CARDS
PRINT RUN DECK
PARAMETER ESTIMATES
(4 data cards)
PRINT RUN DECK
STOP GADS CARDS.
```

As previously, the "PRINT" statements are optional.

SECTION 3

RUN DECK

3.1 CONVENTIONS

In the explanations that follow, the following simplifying conventions are adopted:

a. Control cards pertaining to GADS are referenced by a number prefixed by the letter G. Hence, G13, for example, refers the reader to the 13th GADS control card. This number is in no way related to the program but is used to assist in the documentation. (Control cards are identified in the program solely by their mnemonics punched in columns 1 through 18.) Moreover, the expression "G13-2", for example, refers the reader to the second card under G13.

b. Standard FORTRAN format specifications are included to aid in the description of the cards.

c. All hollerith symbols are left-justified and have six characters.

d. All hollerith directives are 18 characters. The first word is left-justified; thereafter, words are separated by one space.

3.2 DEFINITIONS

3.2.1 VECTORS, SETS, ELEMENTS

Lists of symbols may be ordered or unordered; they are referred to as vectors and sets, respectively. Lists of quantities are always vectors and the entries in a list are elements; that is, vectors and sets are made up of elements. (NOTE: Cartesian vectors are vectors of dimension 3. In this document, a vector is not assumed to be cartesian unless stated explicitly.)

3.2.2 SPECS

The acronym "specs" will refer to data entered according to one of the specific rules stated below for some specific subject. Examples of subjects are sensors, complex of sensors, SC-4020 display requests, etc. Hence, the expression "sensor specs under G13-2", for example, refers to the sensor-related vectors and sets punched in card G13-2.

3.2.3 OBSERVABLE

Observable quantities are those to which one might wish to fit a theoretical curve. Typical observables are the output values obtained from sensors and are not limited to on-board sensors. For example, the amplitude of telemetry signals are sometimes available in recorded form. This amplitude is sensed by the ground receiver antenna and is a function of several factors which include the distance and relative orientations of the transmitting and receiving antennas. For some cases, it is possible to infer some knowledge of the spacecraft's attitude from this amplitude variation. Receivers can sometimes compensate for variation of signal strength by incorporating an automatic gain control (AGC) technique. Refer to page 188, NASA-TN-D608. An observable signal, then, is any signal available to the computer which is helpful in attitude determination.

3.2.4 COMPLEX OF SENSORS

The GADS system is designed to fit predicted (theoretical) curves to the observed outputs of several sensors at once. That is, the GADS system is a vector differential correction program. It is implied, therefore, that there is a way to define the member sensors (components) in a given vector differential correction problem. These participating sensors are referred to as a Complex of Sensors. In GADS, each sensor complex is defined at object time, as explained below. As many as eight sensor complexes may be defined, each with arbitrarily chosen sensors and each with its own set of options.

A complex of sensors, therefore, constitutes a vector differential correction problem complete with its own components, parameters, displays, and other options. Thus the control card COMPLEX OF SENSORS is to be regarded as a GADS executable statement. If more than one such complex is defined, the order of execution is explained below (see card G14a-1).

If a certain complex of sensors results in successful convergence, the newly refined parameters are transmitted to the next complex as initial estimates. This takes place in subroutine GADSPT. It is evident that a series of intermediate parameter refinements can be accomplished with the use of this type of control card. Parameter cascading, as it is sometimes called, is discussed in Reference 1.

When a given complex fails to obtain convergence, the others are not necessarily affected. Hence, one may employ these complexes as auxiliary operations for various other reasons. For example: a.) alternate methods of solutions may be tried during the same run, b.) some non-essential parameters may be desired without jeopardizing the main solution, and c.) special displays can be generated by dummy complexes. In the last example, the display option is the main interest. The complex would be defined so as to iterate only once. The most important factor to keep in mind in defining a sensor complex is to insure that it leads to a "well-posed" system of normal equations. For example, a complex containing only symmetry axis sensors leads to singular equations for balanced spinning spacecraft. Refer to Reference 1, page 80.

3.2.5 PARAMETERS

A quantity is defined to be a parameter if it is to be refined by the method of least squares differential correction. The set of parameters may contain orientation angles (right ascension, declination, inclination), spin rates, spin periods, calibration constants, and configuration constants. A quantity will be refined by differential correction if it is assigned a symbol and its symbol appears in the PARAMETER NAMES specs, G5-1, 2, 3, 4. That is, a quantity becomes a parameter if it is assigned a unique symbol and this symbol appears only once in G5-1, 2, 3, 4.

3.2.6 CALIBRATION CONSTANTS

Let the output signal, f, of a certain sensor be described by the equation:

$$f = \mathscr{F}(x, a_1, a_2, a_3, \ldots) \tag{1}$$

where $\mathscr{F}$ is some function of x and a_1, a_2, a_3... The latter are constants determined by the hardware and x represents that part of the space environment upon which the instrument operates. (For example, x may be the intensity of solar radiation reaching the instrument.) Then the constants a_1, a_2, a_3, ... are termed calibration constants. Typical calibration constants are zero level bias and slope. For a solar sensor with a linear response, equation (1) would be:

$$f = a_1 + a_2 x. \tag{2}$$

Thus a_1 is the zero level bias and a_2 is the slope.

3.2.7 MOUNTING CONSTANTS

Configuration or mounting constants are defined as follows: Consider the quantity x discussed in the foregoing paragraph. x is obviously a function of the spacecraft orientation with respect to the space environment, say solar radiation, S. Hence, x will depend on the vector quantity

$$E \cdot S \tag{3}$$

where E is a 3 x 3 coordinate transformation from the space environment system to vehicle coordinates. Moreover, x depends on the orientation of the sensor sensitive axis (or surface) with respect to vehicle coordinates. That is, x also depends on the sensor mounting. Hence,

$$x = K \cdot E \cdot S \tag{4}$$

where K is a cartesian vector describing the sensor mounting with respect to vehicle coordinates. Then the components of K are called mounting constants. When a vehicle possesses moving parts, mounting constants may also be defined. Refer below to cards G11-1, 1 and G13-1 through 7.

3.3 RULES

Specs are discussed individually for each card. Rules given under a certain card to not necessarily carry over to other cards. Since certain rules and restrictions appear often, they warrant the following assignment of names:

3.3.1 PACK RULE

When a list is to be punched without skipping fields, it is said to obey the PACK rule. This rule does not imply that the list must be totally filled. It does mean, however, that the first blank field indicates the end of the list.

3.3.2 ORDER RULE

If the order of appearance of the entries in a list is relevant, the list is said to have an ORDER rule and the list is a vector. Otherwise, it is understood that elements of a list may be permuted.

3.3.3 CORRESPONDENCE RULE

When elements of one vector are to be related to those of another the association is achieved by the element number. This rule is called the CORRESPONDENCE rule.

3.4 ORDER

The run deck for the UNIVAC 1108 EXEC VIII system is arranged as follows:

Card No.	Contents and Comments	
1	@ RUN	
2	@ ASG,T GADS,T,NNNN	(GADS program)
3	@ASG,T USER,T,MMMM	(user program)
4	@ASG,T H,T	(display tape)
5	@ASG,T DATA1,T,I	(optional)
6	@ASG,T OUTPUT,T,J	(optional)
7	@ASG,T ORBIT, T,K	(optional)
8	@ASG,T ORBIT,T,K	(optional)
9	@ASG,T ORBIT,T,K	(optional)
10	@REWIND USER.	
11	@REWIND GADS.	
12	@COPIN USER.	
13	@COPIN GADS.	
14	@FREE GADS.	
15	@FREE USER.	

3.4.1 BASIC RUN DECK

The basic GADS run deck is shown in Figure 3-1. The first card must always be START GADS CARDS; the last card, STOP GADS CARDS, as shown. Cards G3 through G18 define the attitude determination calculations. Their images are maintained in core and are scanned each time module GADSIN is called. Note that the contents of cards G3 through G18 may be loaded into the array RUNDEK by means of DATA statements. In such a case, the control cards G2 through G18 are not needed. The physical deck then becomes significantly reduced.

The cards are considered in groups. Groups consist of one or more control cards together with their list, data, or specification cards. The number of cards following each control card is indicated.

Card No.	Contents and Comments	
G1	START GADS CARDS (no data cards)	(Required)
G2	GADS RUN DECK (all the following cards through G18)	(Optional)
G3	PARAMETER CALIBRATIONS (2 data cards)	(Optional)
G4	PARAMETER MOUNTS (2 data cards)	(Optional)
G5	PARAMETER NAMES (4 data cards)	(Optional)
G6	PARAMETER ESTIMATES (4 data cards)	(Required)
G7	PARAMETER ACCURACY (4 data cards)	(Optional)
G8	PARAMETER CONVERGENCE (4 data cards)	(Optional)
G9	PERTURBATIONS (4 data cards)	(Optional)
G10	CONSTRAINT OF MOTION (5 data cards)	(Optional)
G11	AUXILIARY FUNCTIONS (2 data cards)	(Optional)
G12	AUXILIARY ESTIMATES (2 data cards)	(Optional)
G13	OBSERVABLE (SENSOR) (4 data cards or more)	(At least one is required)

Figure 3-1. GADS Run Deck (Sheet 1 of 2)

There may be as many as sixteen OBSERVABLE (SENSOR) groups.

G14a	COMPLEX OF SENSORS (7 data cards)	(At least one is required)
G14b	CARRY AUXILIARY FUNCTIONS (2 data cards)	(Optional)
G14c	DIFFERENTIATE NUMERICALLY (4 data cards)	(Optional)
G14d	CALIBRATE SENSORS (2 data cards)	(Optional)
G14e	CONSTRAINT (2 data cards)	(Optional)
G14f	NUMERICALLY INTEGRATE (2 data cards)	(Optional)

There may be as many as eight groups like 14a through 14f. That is, up to eight sensor complexes may be defined.

G15	DISPLAYS (6 or more data cards)	(Optional)
G16	PROBLEM DEFINITION (1 data card)	(Optional)
G17	USER IDENTIFICATION (1 data card)	(Optional)
G18	STOP (no data cards)	(Required)
G19	STOP GADS CARDS (no data cards)	(Required)

Figure 3-1. GADS Run Deck (Sheet 2 of 2)

3.4.2 DETAILED DESCRIPTION OF RUN DECK

The cards are discussed in detail below. Hollerith inputs are shown in capitals and quotations and are left-justified. Hollerith phrases contain one space between words. When in doubt, refer to the data statements in module GADSIN.

The formats are designed to give a good impression to the person inspecting a listing of the cards. It will be noted that the essentials of an attitude problem are readily discernible. To obtain a listing, see below.

Card No.	Card Col.	Format	Variable	Commentary
G1				Required. This must always be the first card read by GADS. The card input stream will continue to be read by module GADSIN until either the stream is exhausted or card G19 is encountered.
	1-18	3A6	CARD	This field contains "START GADS CARDS".
	19-80			Ignored.
G2				This control is needed when the GADS run deck is to be loaded into the scanning area RUNDEK.
	1-18	3A6	CARD	This field contains "GADS RUN DECK". The card stream will then be loaded until a "STOP" is encountered.
	19-80			Ignored.
G3		3A6		Optional. This card enables the user to request the refinement of calibration constants by least square differential correction. That is, this card is used to declare certain calibration constants as parameters.

Card No.	Card Col.	Format	Variable	Commentary
	1-18	3A6	WORK	This field contains "PARAMETER CALIBRATIONS"
G3-1				Required if G3 appears.
	1-6	A6	CALIBP1	This element is the name (symbol) of a parameter. Hence, this element must also appear in the set discussed under G5-1, 2, 3, 4.
	7	1X		Ignored.
	8-13	A6	CALIBP2	This element is the symbol of a sensor. Hence, this element must also appear in Col. 1-6 of a card of the type G13-1.
	14, 15	I2	KCPS	This element is an integer from 1 through 6, inclusively.
	16-20	5X		Ignored.
				NOTE 1: The vector {CALIBP1, CALIBP2, KCPS} instructs the GADS system to refine the KCPS calibration constant belonging to the CALIBP2 sensor and to identify this parameter as indicated in element CALIBP1.
	21-40			Same as columns 1-20.
	41-60			Same as columns 1-20.
	61-80			Same as columns 1-20.
G3-2				Same as card G3-1.

Card No.	Card Col.	Format	Variable	Commentary
				NOTE 2: While each vector {CALIBP1, CALIBP2, KCPS} obviously obeys the ORDER and PACK rules, the 4 available fields for each such vector do not obey either rule. For example, if a certain parameter has been determined from previous runs, its associated data appearing on these cards may be deleted without rearranging the remaining punched data. A total of eight arbitrarily chosen calibration constants may be declared as parameters.
G4				Optional. This card enables the user to request the refinement of configuration or mounting constants. That is, this card is used to declare certain mounting constants as parameters. Note the similarity with card G3.
	1-18	3A6	WORK	This field contains "PARAMETER MOUNTS".
	19-80			Ignored.
G4-1				Required if G4 appears.
	1-6	A6	ORIENP1	This element is the symbol of a parameter. Hence, this element must appear in the set discussed under G5-1, 2, 3, 4.
	7	1X		Ignored.
	8-13	A6	ORIENP2	This element is the symbol of a sensor. Hence, it must also appear in Col. 1-6 of a card of the type G13-1.

Card No.	Card Col.	Format	Variable	Commentary
	14, 15	I2	KOPS	This element is an integer from 1 through 6, inclusively.
	16-20	5X		Ignored.
				NOTE: The vector {ORIENP1, ORIENP2, KOPS} instructs the GADS system to refine the KOPS mounting constant belonging to the ORIENP2 sensor and to identify this parameter as indicated by element ORIENP1.
	21-40			Same as columns 1-20.
	41-60			Same as columns 1-20.
	60-80			Same as columns 1-20.
G4-2				Same as G4-1.
				Consult Note 2.
G5		3A6		Required. This card introduces the set of parameter symbols. It must not precede cards of the type G3 or G4.
	1-18	3A6	WORK	This field contains "PARAMETER NAMES".
G5-1				Required. This card contains the names of the first eight parameters.
	1-6	A6	PARAMS	The name of the first parameter.
	7-10	4X		Ignored.
	11-16	A6	PARAMS	The name of the second parameter, etc.
G5-2		8(A6, 4X)		This card contains the names of the 9th through 16th parameters.
G5-3		8(A6, 4X)		This card contains the names of the 17th through 24th parameters.
G5-4		8(A6, 4X)		This card contains the names of the 25th through 32nd parameters.

Card No.	Card Col.	Formal	Variable	Commentary
				NOTE 3: When the GADS system is employed as a general purpose differential correction loop, the user may define his own parameters and arrange them in any order. In so doing, he must make maximum use of the run deck instructions; i.e., he may not capitalize on implicit data. This type of application is explained separately. Normally, however, implicit data will be understood. The first fourteen parameters, for example, can be implicit. They are required in the determination of the attitude of space vehicles. In this sort of application one may ignore the first 14 parameter names and begin punching the symbols starting with the 15th field. These symbols are those assigned to calibration and mounting constants which appear in cards G3-1, 2 and G4-1, 2. In addition, other parameter names may be included as explained in Section 2. Of course, it is important to avoid duplication of symbols. The following symbols are reserved: 1. "ALPHA" - right ascension, angular momentum. 2. "DELTA" - declination, angular momentum. 3. "PHIO" - first Euler angle. 4. "THETAO" - second Euler angle.

Card No.	Card Col.	Format	Variable	Commentary
				5. "PSIO" - third Euler angle.
				6. "PHIDOT" - first Euler rate.
				7. "PSIDOT" - third Euler rate.
				8. "AAA" - first moment of inertia.
				9. "BBB" - second moment of inertia.
				10. "CCC" - third moment of inertia.
				11. "PO" - x component of Ω.
				12. "QO" - y component of Ω.
				13. "RO - z component of Ω.
				14. "TO" - origin of time for elliptic functions (Reference 1).
G6				Required. This card serves to introduce initial estimates of the parameters.
	1-18	3A6	WORK	This field contains "PARAMETER ESTIMATES".
	19-80			Ignored.
G6-1			UEST	Required. This is the first of four cards which must appear following G6.
	1-10	E10.5	UEST	This element is the estimated value of the first parameter.
	11-20	E10.5	UEST	The value of the second parameter.
	21-30	E10.5	UEST	The value of the third parameter.
. . . .				Etc. . . . .

Card No.	Card Col.	Format	Variable	Commentary
G6-2		8E10.5	UEST	This card contains the estimated values of the 9th through 16th parameters.
G6-3		8E10.5	UEST	This card contains the estimated values of the 17th through 24th parameters.
G6-4		8E10.5	UEST	This card contains the estimated values of the 25th through 32nd parameters.
				NOTE 4: The CORRESPONDENCE rule relates the elements of cards G5-1, 2, 3, 4 and G6-1, 2, 3, 4.
G7		3A6		Optional. This card serves to introduce data concerning the accuracy (not precision) with which the aforementioned parameter estimates are given. Not ready.
	1-18		WORK	This field contains "PARAMETER ACCURACY".
	19-80			Ignored.
G7-1				This is the first of four cards which must appear if G7 appears. The elements in cards G7-1, 2, 3, 4 are punched exactly as those in cards G6-1, 2, 3, 4. When this option is not exercised, internal values are employed.
G7-2		8E10.5	UACC	See preceding commentary.
G7-3		8E10.5	UACC	See preceding commentary.
G7-4		8E10.5	UACC	See preceding commentary.
G8		3A6		Optional. This card serves to introduce the convergence criteria for the parameters.

Card No.	Card Col.	Format	Variable	Commentary
	1-18	3A6	WORK	This field should contain "PARAMETER CONVERGENCE".
	19-80			Ignored.
G8-1		8E10.5	EPS	This is the first of four cards which must appear if G8 appears. G8-1, 2, 3, 4 are punched exactly as those in cards G6-1, 2, 3, 4.
G8-2		8E10.5	EPS	See preceding commentary.
G8-3		8E10.5	EPS	See preceding commentary.
G8-4		8E10.5	EPS	See preceding commentary. When this option is not exercised, internal values are used.
G9				Optional. This card serves to introduce parameter perturbations which are employed in numerical differentiation.
	1-18	3A6	WORK	This field contains "PERTURBATIONS".
	19-80			Ignored.
G9-1	1-8	E8.3	UCUTS	Perturbation (small number) used to approximate partial derivatives by the method of "false position". Perturbed functions are computed on both sides of center. Let d= UCUTS(i) and let N be an integer (0<N<5). Then GADS will compute the predicted sensor output f for the arguments: U, U+d, U-d, U-2d, U+2d,... U-Nd, U+Nd, where U represents the set of system parameters. The partial derivative of f with respect to U(I) can then be estimated by a formula. See statement 1001, module GADSGC. This is the first of four cards which must appear if G9 appears. These quantities are related to those of G5-1, 2, 3, 4 by the CORRESPONDENCE rule.

Card No.	Card Col.	Format	Variable	Commentary
	9,10	I2	LINKUH	This is the value of the integer N discussed above. At this writing, N should be 1; values greater than 1 are treated as 1. See module GADSGC.
	11-20			Same as columns 1-10.
	.....			....
	71-80			Same as columns 1-10.
G9-2				See preceding commentary.
G9-3				See preceding commentary.
G9-4				See preceding commentary. When this option is not exercised, internal values are used.
G10		3A6		Optional. This card serves to introduce an equation of constraint. Not ready.
	1-18	3A6	WORK	This field contains "CONSTRAINT OF MOTION".
	19-80			Ignored.
G10-1		2(A6,4X),I2		This is the first of five data cards which must follow G10.
	1-6	A6	HTITLE	This element is the name or symbol assigned to the constraint.
	7-10	4X		Ignored.
	11-16	A6	HCODE	This element is the code of the constraint.
	17-20	4X		Ignored.
	21,22	I2	NUMHF	This element is an integer in the range 1-8, inclusively, or may be left blank. (There is no purpose for this field at this writing.)
	23-80			Ignored.

Card No.	Card Col.	Format	Variable	Commentary
G10-2		8I10		This is the first of four cards which provide information about the derivatives of the constraint relation. Note that the elements contained in cards G10-2, 3, 4, 5 are related to the elements of cards G5-1, 2, 3, 4 according to the CORRESPONDENCE rule.
	1-10	I10	NWORK	This element is the number of the constraint subroutine which computes the derivative with respect to the first parameter. It may be left blank when a standard constraint is in use.
	11-20	I10	NWORK	Same as above with respect to the second parameter.
	21-30	I10	NWORK	As above, for the third parameter.
	31-40	I10	NWORK	As above, for the fourth parameter.
	41-50	I10	NWORK	As above, for the fifth parameter.
	51-60	I10	NWORK	As above, for the sixth parameter.
	61-70	I10	NWORK	As above, for the seventh parameter.
	71-80	I10	NWORK	As above, for the eighth parameter.
G10-3				Same as card G10-2 but for parameters 9-16.
G10-4				Same as card G10-2 but for parameters 17-24.
G10-5				Same as card G10-2 but for parameters 25-32.

Card No.	Card Col.	Format	Variable	Commentary
				NOTE 5: The use of CONSTRAINT OF MOTION option implies the use of Lagrange multipliers. The user must then provide unique symbols for each multiplier in the parameter array, cards G5-1, 2, 3, 4. The Lagrange multipliers must appear after all other parameters beginning in the LAGRAN+1st cell. The variable LAGRAN is presently set at 24 but may be changed if desired.
G11				Not used.
G12				Not used.
G13				At least one of these cards is required. By means of this and the next few cards, the user defines a sensor or, at least, its observed output. Hence, this is a "declarative" GADS statement.
	1-18	3A6	CARD	This field contains "OBSERVABLE (SENSOR)".
	19-80			Ignored.
G13-1	1-6	A6	OTITLE	Hollerith sensor name, arbitrary.
	7	1X		
	8-13	A6	OCODE1	Hollerith "sensor operand" type, must be one of the following:
				"SOLAR" - for all solar sensors
				"MAGNET" - for all magnetometers
				"HORIZN - for all horizon scanners. Not ready.

Card No.	Card Col.	Format	Variable	Commentary
				"GROUND", "LUNAR", "STELLR", "PLASMA" are suggestions for future programming. NOTE: The purpose of OCODE1 is to furnish module GADSSO with the variable JSCODE which selects the appropriate orbit vector. Thus, JSCODE=1, selects the sun vector; 2, selects the magnetic field; 3, the orbital radius vector, etc.
	14	1X		
	15-20	A6	OCODE 2	Hollerith sensor output function type, must be either: "COSINE" - for all cosine sensors or "TUNED" - for tuned oscillator. Not ready. NOTE: OCODE2 selects the appropriate sensor output module. For example, "COSINE" results in the use of module GADFO1, "TUNED", in module GADFO2, etc.

Card No.	Card Col.	Format	Variable	Commentary
	21	1X		
	22-25	O4	NUMF	Octal function code. When non-zero, NUMF overrides OCODE2. This feature is provided for use in special problems. Setting NUMF=1, 2, 3... causes the selection of modules GADFO1, 2, 3..., respectively. Normally the user should rely on OCODE2 and leave NUMF blank.
	26-30	I5	LOCATE	Relative location of data in array COMM.
	31-35	I5	LENGTH	Number of data points.
	36-55	3I5	ii, jj, kk	The variables ii, jj, kk are for the user's convenience in case he needs them in the module SENSOR.
	56-80	25X		Ignored.
G13-2				Optional. This card is used to introduce derivative codes. In its absence, the standard derivative codes are used. This control card must be followed by exactly one data card obeying the PACK rule. Up to four such pairs of cards may appear at this point.
	1-18	3A6	CARD	This field contains "DERIVATIVES."
	19-80			Ignored.
G13-3	1-6	A6	WORK	Hollerith name of a system parameter. This name must appear in the list entered under card G5.
	7-10	I4	NWORK	Derivative number in the range 25, 26, ... 40. This code selects a module of the type GADG25, 26, ... 40, respectively.

Card No.	Card Col.	Format	Variable	Commentary
	11-20	A6, I4		Same as columns 1-10.
	21-30	A6, I4		Same as columns 1-10.
	31-40	A6, I4		Same as columns 1-10.
	41-50	A6, I4		Same as columns 1-10.
	51-60	A6, I4		Same as columns 1-10.
	61-70	A6, I4		Same as columns 1-10.
	71-80	A6, I4		Same as columns 1-10.
G13-4				Optional, same as G13-2.
G13-5				Two more pairs like G13-2 and G13-3 may appear (a total of 4). Scanning of these data cards stops when the first blank hollerith field is encountered, when 4 cards have been processed, or when the control card "DERIVATIVES" fails to appear, whichever occurs first.
G13-6	1-3	I3	ACHAR	Character selector code for plotting raw data points. See Table 4 and page II-81, Reference 10.
	4-6	I3	FCHAR(,1)	Character selector code for plotting predicted sensor output. See same reference as above.
	7-9	I3	FCHAR(,2)	Character selector code for plotting calibrated predicted sensor output.
	10	1X		
	11-20	E10.5	SXSPAN	Horizontal span of one display frame, in milliseconds. To avoid excessive output, SXSPAN should not be too small.

Card No.	Card Col.	Format	Variable	Commentary
	21, 20	E10.5	PDENS	Horizontal span between dense predicted function points, milliseconds. To avoid excessive output, PDENS must also not be too small. For an example of a "dense" plot, see Figures 4, 5, and Reference 1.
	21-30			Lower limit of display, engineering units.
	31-40			Upper limit of display, engineering units.
	41-50			Lower limit of display when displaying calibrated sensor output, telemetry counts.
	51-60			Upper limit of same. NOTE: The variables SENMIN and SENMAX correspond to YB and YT in module GRID1V. Also refer to page II-11, Reference 10.
	61-80			Ignored.
G13-7	1-6	A6	OCODE4	Hollerith sensor mounting type. This variable determines geometry (mounting)-dependent calculations in modules of the type GADSR1, 2, etc. Also note that these modules are called by GADSSR. This field may contain one of the following hollerith symbols: "FIXEDV" - code for a fixed unit vector. This describes the vast majority of sensor mountings "AXIS V" - n.r., code for a sensor mounted on an axis, "SPECIA" - n.r., code for special type of mounting.
	7-10	4X		Ignored.

Card No.	Card Col.	Format	Variable	Commentary
	11-70	6E10.5	OBMOUN	These numbers are the "raw mounting constants". In the case of a "FIXEDV" type of sensor, the first two numbers are Φ and Θ, respectively. These angles are illustrated in Figure 16. The remaining numbers are ignored but are provided for possible future use. Conceivably the user may wish to deal with a sensor mounting defying definition by merely six numbers. In such a case there is nothing to keep him from providing more constants separately, for example, through a COMMON area to a module of the type GADF01. The main purpose and advantages of OBMOUN are: 1) any of these constants can be refined by differential correction by declaring them as parameters. See card G4. 2) These raw constants are automatically processed to produce the equivalent direction cosines. See module GADSR1.
	71-80	10X		Ignored.
G13-8	1-6	A6	OCODE3	Hollerith sensor calibration function type. This variable determines the conversion of engineering units to telemetry counts. When the user so specifies, the results obtained in modules like GADF01 can be transformed to telemetry counts by means of modules of the type GADC01. (Alternatively, the observed data can be calibrated. See module GADC04.) Thus OCODE3 is used to select these calibrating modules: "POLYN." - results in a polynomial calibration, module GADC01, "FURIER", "OSCIL.", "POLY. D" - results in a polynomial calibration, module GADC04.

Card No.	Card Col.	Format	Variable	Commentary
	7-10	4X		Ignored.
	11-70	6E10.5	OBCOEF	These numbers are the "raw calibration constants". In the final analysis, the significance of these numbers depends on their usage in the aforementioned modules. The comments concerning OBMOUN, above, are applicable to OBCOEF. Like those quantities, OBCOEF can be adjusted by differential correction using card G3.
G13-9	1-10	E10.5	OBWGHT	This is a constant weighting factor that will be applied to all observations derived from the present sensor. When the user wishes to apply a variable weighting function, consult cards G14-2, 3.
	11-80			Ignored.
G14				At least one of these cards and its data cards must appear. By means of this group, the user defines a "sensor complex". Hence, this is an "executable" GADS statement and cannot appear before any group of the type G13.
	1-18	3A6	CARD	This field contains "COMPLEX OF SENSORS".
	19-80			Ignored.
G14-1	1-6	A6	CTITLS(, 1)	Hollerith sensor complex name, arbitrary.
	7	1X		Ignored.
	8-13	A6	CTITLS(, 2)	Hollerith name of "next executable" sensor complex if the present complex obtains convergence in differential correction. If this field is blank, the next complex is simply the next one by order of declaration. If this is the last complex, however, all differential correction is assumed completed.

Card No.	Card Col.	Format	Variable	Commentary
	14	1X		Ignored.
	15-20	A6	CTITLS(, 3)	Hollerith name of "next executable" sensor complex if the present complex fails in convergence. If this field is blank, no error alternative is assumed and differential correction stops. A "RETURN" will have the same effect.
	21	1X		Ignored.
	22-27	A6	CTITLS(, 4)	Hollerith special purpose code, normally blank, may contain: "DUMPS" - causes raw data, sensor data to be printed whenever the sensor complex fails to obtain convergence. "OUTPUT" - causes this complex to be used for generating output. See GADSAT.
	28	1X		Ignored.
	29-34	A6	CTITLS(, 5)	Hollerith special purpose code, normally blank, may contain: "STATIS" - causes calculation of statistical measures of convidence. See module GADSST.
	35	1X		Ignored.
	36-40	I5	IQUITS	Maximum number of iterations.
	41-45	I5	NUMC	"Erase" outlier flag, normally zero. NUMC=1 will cause outliers to be "erased" from the input data in COMM. See module GADSLS notes.
	46-50	I5	NTRYS	Maximum number of outlier detection trials, normally zero.
	51-55	I5	IGNORX	Ignore present complex flag, normally zero. When IGNORX=1, the present complex will be ignored by module GADSLS.

Card No.	Card Col.	Format	Variable	Commentary
	55-60	I5	MOTNS	Type of spacecraft motion. The types of motion already programmed are MOTION=1: simple spin, MOTION=2: balanced Euler motion, MOTION=3: non-balanced motion, MOTION=4, 5...: not completed.
	61-65	I5	-	Ignored.
	65-70	5X		Ignored.
	71-80	E10.5	CTOLER	Criterion for an outlier. After a convergence cycle, data points outside a "band" of $\pm$ CTOLER* SDEV are considered outliers. SDEV is the unbiased root-mean deviation of the fit (the weighted residuals).
G14-2	1-6	A6	WORK	Hollerith sensor name. This field must contain one of the OTITLE symbols. See card G13-1.
	7-10	I4	NWORK	Normally blank or zero. When non-zero, this field causes the sensor weighting function to be calculated by means of the corresponding module of the type GADW01, 2...
	11-20			Same as 1-10.
	21-30			Same as 1-10.
	31-40			Same as 1-10.
	.....	.....		....
	71-80			Same as 1-10.
				NOTE: Card G14-2 obeys the PACK rule. However, the next card must be present even if totally blank.
G14-3	1-80	8(A6, I4)		Same as card G14-2.
G14-4	1-6	A6	WORK	Hollerith parameter name. The contents of this field must have appeared somewhere in cards G5-1, 2, 3, 4. The parameter named here will be a "participating parameter".

Card No.	Card Col.	Format	Variable	Commentary
	7-10	4X		Ignored.
	11-16	A6		Same as 1-6.
	17-20	4X		Ignored.
	21-30	A6, 4X		Same as 1-10.
	.....	.....		....
	71-80	A6, 4X		Same as 1-10.
				NOTE: Card G14-4 also obeys the PACK rule. However, the next card must also appear even if blank.
G14-5	1-80			Same as G14-4.
G14b				Optional. This card is used to cause the present sensor complex to carry certain auxiliary functions, i.e., to call certain modules of the type GADA01. In the absence of auxiliary functions, the user should either use controls G14f or replace module GADSAL with a version which computes the attitude matrix.
	1-18	3A6	CARD	This field contains "CARRY AUXILIARY FUNCTIONS".
	19-80			Ignored.
G14b-1	1-6	A6	WORK	Hollerith symbol, must be one of the following: "EULANG" - for computing the Euler angles, uses module GADA01; "R. P. Y." - for roll, pitch, yaw, uses module GADA02 (n. r.); "AUX3" - spare, "AUX4" - spare, "AUX5" - simplified version of Euler angles, uses module GADA05. "AUX6", "AUX7", "AUX8" - spares.
	7-10	4X		Ignored.
	11-20	A6, 4X	WORK	Same as 1-10.

Card No.	Card Col.	Format	Variable	Commentary
				NOTE: GADS will call the modules requested in the order of declaration. The outputs of these modules will be stored sequentially in array AUXF. The number of output functions generated by a certain module, for example, GADA03, is determined by the contents of NAUXDI(3). If the user programs one of these modules, he should, therefore, insure that NAUXDI contains the correct constants. See block data module GADSDA.
	21-30	A6,4X		Same as 1-10.
	.....	.....		....
	71-80	A6,4X		Same as 1-10.
G14b-2				Same as G14b-1. Both of the preceding cards must appear if G14b appears. Both also obey the PACK rule.
G14c				Optional. This card is used to request that partial derivatives be computed by the numerical perturbation (false position) method for certain parameters designated below. A card of the type G9 must precede this group.
	1-18	3A6	CARD	This field contains "DIFFERENTIATE NUMERICALLY".
	19-80			Ignored.
G14c-1	1-6	A6	WORK	Hollerith name of a parameter declared in card G5-1, 2, 3, or 4.
	7-10	4X		Ignored.
	11-20	A6,4X	WORK	Same as 1-10.
	21-30	A6,4X	WORK	Same as 1-10.
	.....	.....	.....	....
	71-80	A6,4X	WORK	Same as 1-10.

Card No.	Card Col.	Format	Variable	Commentary
G14c-2	1-80			Same as G14c-1.
G14c-3	1-80			Same as G14c-1.
G14c-4	1-80			Same as G14c-1.
				NOTE: The four preceding cards must all appear if G14c appears and they obey the PACK rule.
G14d				Optional. This card is used to request the calibration of certain sensors during the computation of predicted output functions.
	1-18	3A6	CARD	This field contains "CALIBRATE SENSORS".
	19-80			Ignored.
G14d-1	1-6	A6	WORK	Hollerith name of a sensor. This symbol must be one of those entered into OTITLE, card G13-1.
	7-10	4X		Ignored.
	11-20			Same as 1-10.
	.....			....
	71-80			Same as 1-10.
G14d-2	1-80			Same as card G14d-1. The two preceding cards are required if G14d appears. The PACK rule is in effect.
G14e				Not programmed.
G14f				Optional. This card is similar to G14b. The purpose is to request integration of differential equations by means of module ADAMS.
	1-18	3A6	CARD	This field contains "NUMERICALLY INTEGRATE".
	19-80			Ignored.

Card No.	Card Col.	Format	Variable	Commentary
G14f-1	1-6	A6	WORK	Hollerith symbol, must be one of the following: "EULDEQ" - to cause integration of Euler equations of motion using module GADD01; "RPYDEQ" - to cause roll, pitch and yaw equations of motion (n.r.); "DEQ3", "DEQ4", "DEQ5", "DEQ6", "DEQ7", "DEQ8" - spares.
	7-10			Ignored.
	11-20	A6,4X	WORK	Same as 1-10.
	.....	.....	.....	....
	71-80	A6,4X	WORK	Same as 1-10.
				Same as card G14e-1. NOTE: The note at the end of card G14b-2 also applies here.
				NOTE: Cards of the type G14b may appear before or after cards of the type G14e, depending on the requirements of the user.
G15				Optional. This card enables the user to request displays.
	1-18	3A6	CARD	This field contains "DISPLAYS".
	19-80			Ignored.
G15-1	1-7	A6,1X	WORK	Hollerith symbol identifying the subject matter for display, must be one of the following: "COMPLEX" - prepare to display a sensor complex; "ORBIT" - (n.r.) display orbit parameters; "DETAIL" - prepare to load display controls.
	8-10			Ignored.

Assuming that the previous entry was "COMPLEX", the card columns are as follows:

Card No.	Card Col.	Format	Variable	Commentary
	11-16	A6	WORK	Hollerith symbol identifying the complex to be displayed.
	17-20			Ignored.

Card No.	Card Col.	Format	Variable	Commentary
	21-26	A6	WORK	Hollerith symbol specifying the level of the display, must be one of the following: "FINALS" - display final results. If outlier detection is done, display after outliers are isolated; "BYSTEP" - display at each iteration of differential correction; "SUPERP" - superimpose all sensor outputs on same picture; "DIFCOR" - display after each time that differential correction converges; i.e., in-between outlier detection trials.
	27-30			Ignored.
	31-36	A6	WORK	Same as above.
	37-40			Ignored.
	41-46	A6	WORK	Same as above.
	47-50			Ignored.
	51-56	A6	WORK	Same as above.
	57-80			Ignored. <u>NOTE:</u> These last four fields are independent and do not obey the PACK, ORDER, or CORRESPONDENCE rules.
G15-2	1-6	A6	WORK	Hollerith symbol, the name of a sensor to be displayed.
	7-10			Ignored.
	11-16	A6		Same as above.
	.....			....
	71-76	A6		Same as above.
G15-3				Same as G15-2. <u>NOTE:</u> Fields in cards G15-2 and G15-3 do not obey the PACK, ORDER, or CORRESPONDENCE rules.

Card No.	Card Col.	Format	Variable	Commentary
G15-4	1-6	A6	WORK	Hollerith symbol identifying the type of data to be displayed, must be one of the following: "APLOT" - plot raw data; "FPLOT" - plot predicted sensor output function, engineering units; "TPLOT" - (n.r.) plot predicted sensor output, telemetry counts; i.e., plot calibrated function; "RPLOT" - (n.r.) plot the residuals; "DENSEF" - display predicted function at intervals determined by the variable PDENS (see card G13-6), as well as the sample times. This allows the user to receive a continuous curve. See Figure 4, Reference 1; "ENVELP" - plot the "one sigma" envelope.
	7-10			Ignored.
	11-16			Same as above.
	.....			....
				NOTE: The preceding six fields do not obey the PACK, ORDER, or CORRESPONDENCE rules.
	57-80			Ignored.

Assuming that the entry in card G15-1 was "DETAIL", the cards are as follows:

G15-1a	7-80			Ignored.
G15-2a	1-80			Ignored.
G15-3a	1-80	8I10	LDISP	Module GRID1V controls. The elements of LDISP(I), I=1, 2, 3...8 are:

I	Variable and Comment
1:	L
2:	N
3:	M
4:	I
5:	J, where these variables are those shown on page II-11, Reference 10

Card No.	Card Col.	Format	Variable	Commentary
				6: IDENS - maximum number of predicted sensor output functions to interpolate between any two observations when using the "DENSEF" option (see card G15-4) 7,8: Not used
G15-4a	1-80	8E10.5	FDISP	Module GRID1V controls. The elements of FDISP(I), I=1,2,3...8 are: I Variable and Comment 1: DX 2: DY 3-8: Not used These variables are also explained on page II-11, Reference 10.
G15-5a	1-80	8I10	LDISP	Same as card G15-3a except that LDISP(I), I=9,10,...16 are loaded. These elements have the same significance as those of card G15-3a.
G15-6a	1-80	8E10.5	FDISP	Same as card G15-4a except that FDISP(I), I=9,10,...16 are loaded. These elements have the same significance as those of card G15-4a. NOTE: The reason for there being two complete sets of grid control variables is that in module GADSTV there are two internal submodules which may require independent controls. The first set belongs to submodule GRAPH1; the second to GRAPH2. The first module displays sensors separately; the second superimposes all sensors on one picture.
G16				This card is now optional, but may be required in the future when non-Eulerian methods of parametrization are included in the GADS system. See Appendix C, Reference 1.

Card No.	Card Col.	Format	Variable	Commentary
	1-18	3A6	CARD	This field contains "PROBLEM DEFINITION".
	19-80			Ignored.
G16-1	1-36	9A6	WORK	This field determines the type of angles used to describe the motion. At present, this field may contain only: "THIS PROBLEM IS DEFINED IN TERMS OF EULER ANGLES". Strictly speaking, this group G16, G16-1, belongs ahead of group G5.
G17				This optional card is used to enter user identification used by GSFC SC 4020 software.
	1-18	3A6	CARD	This field contains "USER IDENTIFICATION".
	19-80			Ignored.
G17-1	1-36	6A6	USERID	See Reference 10, page II-2.
	37-80			Ignored.
G18				Required. This card is used to stop the scanning of card images, i.e., to halt initialization in module GADSIN.
	1-18	3A6	CARD	This field contains "STOP".
	19-80			Ignored.
G19				Required. This card is used to stop the reading of cards from the input stream. Upon reading this card, module GADSIN reads no more physical cards.
	1-18	3A6	CARD	This field contains "STOP GADS CARDS".
	19-80			Ignored.

3.4.3 SPECIAL PURPOSE CONTROL CARDS

The following control cards may appear only in the input card stream, that is, never in the array RUNDEK.

CARD	CONTENTS AND COMMENTS
G20	START GADS CARDS - must always be the first card in the input control stream.
G21	GADS RUN DECK - instructs the module GADSIN to begin loading cards into the array RUNDEK until a STOP is found. The card images are counted and the count is printed.
G22	PRINT RUN DECK - causes module GADSIN to print the card images in RUNDEK until a STOP is found.
G23	GADS RUN DECK UPDATE - causes the module GADSIN to modify the contents of RUNDEK in a manner described in paragraph 2.10.
G24	STOP UPDATE - causes module GADSIN to terminate updating.
G25	STOP GADS CARDS - causes module GADSIN to terminate reading the input card stream and proceed to scanning of the array RUNDEK. Also causes the flag INRSET to be cleared thus signalling that the card stream has been exhausted. On succeeding calls, therefore, module GADSIN will proceed directly to the scan mode.

SECTION 4

DIAGNOSTICS

4.1 INITIALIZATION DIAGNOSTICS

The diagnostics described below pertain only to the interpretation of the GADS Run Deck. Hence, these diagnostics are useful in achieving self-consistency in a problem definition. They are not sufficient, however, to insure the success of the differential correction problem since an ill-defined least square problem can be self-consistent. In short, these diagnostics bear a similar relationship to the attitude problem as do the FORTRAN compiler diagnostics to a given mathematical problem.

The diagnostics and codes are shown in the first two columns, respectively. The explanation is given in the third column. The FORTRAN statement numbers are included in the parenthesis to aid in understanding module GADSIN.

Diagnostic	Code	Code Clarification and Statement Number
UNPROGRAMMED MATERIAL	10	Refer to display requests (141, 142, 144, 146).
CARD STREAM		Difficulty in physical card input.
	1	First card should be START GADS CARDS (243).
UPDATE CARDS		Difficulty in update cards.
	1	Not used.
	2	Expecting UPDATE card. Not found (219).
	3	Device error reading update control card (221).
	4	Hit EOF while reading same (222).
	5	Device error while reading update (223).
	6	Hit EOF while reading same (224).
	7	Updates cause RUNDEK to be overloaded. Solution: increase the dimension of same using PARAMETER NOFC. (225).
RUN DECK		Difficulties loading GADS Run Deck.
	1	Device error reading card stream (244).
	2	Hit EOF reading card stream. Missing STOP card (245).
	3	Device error reading the "START GADS CARDS" card only (241).
	4	Hit EOF while reading same (242).
	5	Device error while loading run deck into RUNDEK (170).
	6	Hit EOF while loading RUNDEK (171).
	7	Too many cards. Increase the PARAMETER NOCD (148).
	8	Too many card images. Need the STOP card image (176).
	9	You are trying to replace a group of cards in array RUNDEK. The control card image is not found (232).

Diagnostic	Code	Code Clarification and Statement Number
BAD SENSOR DEFINITION		Difficulty interpreting sensor definition cards.
	1	Unknown sensor operand (819).
	2	Implied derivative with respect to unknown parameter (117).
	3	Unknown sensor function type (821).
	4	Unknown sensor calibrating function or operator type (822).
	5	At least two calibration constants or coefficients (162) should be available.
	6	Unknown mounting type (165).
ORDER, CONTROL CARD		GADS control out of order.
	1	CONSTRAINT card should not be placed before PARAMETER NAMES (806).
	2	COMPLEX OF SENSORS should not appear before sensors are defined (807).
	3	Request for "false position" derivative should not appear before COMPLEX OF SENSORS (816).
	4	Attempting to change parameter names too late. Change parameter names before STOP GADS CARDS (808).
	5	CONSTRAINT OF MOTION should not appear before PARAMETER NAMES (839).
	6	A sensor should not be defined before system parameters, i.e., before PARAMETER NAMES (841).
	7	PARAMETER CALIBRATIONS should be in front of PARAMETER NAMES (825).
	8	PARAMETER MOUNTS should be in front of system parameters; i.e., previous to PARAMETER NAMES (826).
	9	Attempting to perform card operations after STOP GADS CARDS (827).
	10	Perturbations should be defined prior to request for DIFFERENTIATE NUMERICALLY (842).

Diagnostic	Code	Code Clarification and Statement Number
MISSPELLING		A symbol is misspelled.
	1	Request unknown parameter under a COMPLEX OF SENSORS card (801).
	2	Request unknown sensor name under same (802, 804).
	3	Request unknown auxiliary module under same (815).
	4	Request unknown differential equations under same (838).
	5	Request unknown constraint module under same (803).
	6	Request calibration of unknown sensor under same (824).
	7	Request a calibration constant belonging to unknown sensor to be a system parameter (905).
	8	Request a calibration constant of a given sensor to be an unknown system parameter.
	9	Same as 7 for mounting constant (915).
	10	Same as 8 for mounting constant (916).
	11	Attempting to request a special derivative module for a parameter not found under PARAMETER NAMES (156).
	12	Not used.
	13	Request numerical (false position) derivative for a parameter not found under PARAMETER NAMES.
	14	Unknown problem definition (100).
ORBITAL SPECS.		Not used.
UNKNOWN CONTROL CARD		Unrecognizable control card.
	1	In RUNDEK array (805).
	2	In card stream (230).

Diagnostic	Code	Code Clarification and Statement Number
DISPLAY SPECS.		Trouble with display request.
	1	The requested subject for the display in unknown (809).
	2	The requested complex of sensors to be displayed has not been defined (810).
	3	The sensor complex cannot be located (811).
	4	The requested level (amount) of display is unknown (812).

4.2 DIAGNOSTICS FROM DIFFERENTIAL CORRECTION MODULE GADSLS

To aid the user in determining the nature of difficulties in differential correction, the differential correction executive module GADSLS provides the following messages:

ERROR MAT. INV.

The matrix inversion module MATINV detects singular normal matrix of coefficients. Assuming that nondegenerate functions are being calculated, the cause is probably redundant parametrization. See Chapter VI, Reference 1.

NOT CONVERGING

The given complex of sensors does not converge within the allotted number of iterations. Check convergence criteria - possibly too stringent. See cards G8-1, 2, 3, 4. Other possible causes are: a) IQUIT is too small (see card G14-1), b) error in the computation of a partial derivative, c) error in the predicted function calculation, d) mismatch in scaling (calibration) between observed and predicted calculations, e) parameters are not independent, and f) initial parameter estimates are poor.

ACC O. F. COEFFIC.

Calculation of normal matrix elements results in accumulator overflow. Since the summation is performed in double precision (module GADSML), the cause is a faulty prediction function or a derivative.

WORK AREA EXCEEDED

The array COMM is too small or argument LOCEND is not properly set. See paragraph 2.6.

EXCESSIVE XLAMBDA

This refers to Marquardt's x_{λ}. When this quantity is very large, the pure gradient method is being invoked. When this method cannot reduce the squared error, it is necessary to assume that the squared error is already minimized. The cause may be that the parameters are fully adjusted and no further iterations are required. Hence, this diagnostic is not necessarily a cause for alarm. See Notes, module GADSLS.

GRADIENT VANISHES

As the previous diagnostic, this is not a cause for alarm. It may arise under similar circumstances when the "system" has arrived at a local minimum.

SECTION 5

GLOSSARY

<u>Accessory, GADS.</u> A module is considered an accessory to GADS if it is not crucial to the executive modules; i.e., if it is used only at the working level. Hence, an accessory can be replaced by a "do nothing" module if a certain application allows.

<u>Accuracy.</u> The accuracy of a measurement is a measure of how close it is to the "correct" value which is always unknown. This is in contrast to "precision" which is only a measure of the consistency of a result, usually stated in standard deviations. Thus the initial estimates of system parameters should be as <u>accurate</u> as possible. In GADS, the degree of accuracy for the initial estimate of a system parameter ranges between complete certainty and complete uncertainty.

<u>Active Parameter</u>. When a system parameter is being adjusted by differential correction, it is called an "active parameter". A system parameter can be a mounting constant or a calibration constant. When one of these becomes "active", it must be returned to its respective library in order to maintain the library updated. See module GADSPT.

<u>Active Position.</u> The active position of a parameter refers to its position in the normal equations. This position is determined by the order in which parameters are declared for differential correction within a sensor complex; i.e., by the order in which parameters are declared on card G14-4, 5. See Run Deck.

<u>Active Sensor.</u> Like an active parameter, a sensor becomes active when it is declared for participation in a sensor complex during differential correction. See card G14-2, 3, Run Deck.

Auxiliary Coordinate System. This is the intermediate coordinate system to which the attitude angles refer. When using the Euler method of parametrization, the attitude angles are the Euler angles. Also see index, Reference 1.

Auxiliary Function. A function is auxiliary in GADS if it must be computed as an intermediate step in order to predict the attitude and the partial derivatives, and if it is not derived from the environmental variables. The state vector is an example. See notes, module GADSFC.

Backup Parameters. During differential correction, a set of system parameters is maintained as a backup if the said correction is divergent. See note 1, module GADSLS.

Cascading. Differential correction can be performed in stages, each stage dedicated to the refinement of a unique set of system parameters. As improvements of system parameters accumulate from one stage to the next, they are said to cascade. Cascading is desirable when some parameters are known with higher accuracy than others. Thus the latter should be corrected first in order to reduce the risk of wayward differential correction.

Calibration Operator or Function. Because telemetry from artifical satellites seldom provides data in engineering units, it must be converted to these units before residuals can be computed. An alternative is to convert the theoretical functions into telemetry counts. In GADS, both types of conversions are called calibrations. Conversion of engineering units to telemetry counts is unhappily called straight forward calibration; conversion of telemetry counts to engineering units is inverse calibration. See modules GADC01 and GADC04.

Central Limit Theorem. This theorem is used in module GENORM. Consult any comprehensive text on mathematical statistics.

Communication by Interpretation. Several modules have explicit I/O arrays. In the cases of worker modules, the mathematical or physical significance of the array elements is determined by their positions. This type of communication which is called communication by interpretation, has been avoided entirely in the executive modules. Hence, the user need not be concerned with these modules when applying GADS to problems defined in terms other than Euler angles. Only the worker modules need be understood in such instances. See modules GADA01, GADD01, GADSAL.

Complex of Sensors. See Usage.

Cosine Sensor. See notes, module GADF01.

Counts, Telemetry. See Telemetry Counts.

Data Pass. This is, loosely speaking, a collection of data obtained from one spacecraft pass (fly-by) over a given ground station. It normally results in continuous time coverage for as long as the spacecraft transmits, or as long as it is within sight of the station. Sometimes a pass is a programmer cycle.

Derivatives. The derivatives in GADS are those needed to compute the theoretical (or predicted) rate of change of the observational variables (sensor outputs) with respect to the system parameters. This problem often involves intermediate derivatives. See GADF01, GADC01, and GADSR1, for example. Sometimes derivatives must be estimated by numerical methods as in module GADSGC. Derivatives can also be computed as special closed-form functions (as in GADG26, 27, 39 and 40) or they can be obtained by integration. In the last case, the integration may be a quadrature or the solution to a variational equation. Table 2 illustrates the classification of derivatives in GADS.

Differentiation. See Derivatives.

Differentiation, Numerical. See notes, module GADSGC. Refer also to method of false position, regula falsi, and Run Deck, cards G14c.

Do-Nothing Module. This module contains only a return executable statement. The purpose is to relinquish core space. In most applications, several modules in GADS are inactive and can, therefore, be replaced by dummy do-nothing modules.

Engineering Units. Observed data can be thought of in terms of engineering (physical) units such as millioersteds, gammas, volts, or milliamperes. The data is normally received in terms of telemetry counts which are dimensionless. The two are related by a calibration equation or table. See Run Deck, card G13-8, modules GADC01, 4 and Chapter VI, Reference 1.

Environmental Data. See Orbital Data.

Environmental Variable. A physical quantity which is a function of the spacecraft's position. See Orbital Data.

Executive. A module in GADS is an executive if it is not concerned with the context of the problem but merely with the logic of least-squares differential correction.

Explicit I/O Variables. These are variables transmitted via the calling sequence. Note Implicit I/O Variables.

Euler Angles. See Reference 1.

Eulerian Method of Definition or Parametrization. This is the method of using Euler angles to describe the motion.

False Position Method. See Method of False Position.

Fast Access Area. When a key variable is referenced frequently (inside the main data loop), it is placed in a fast access area; i.e., it is placed in named common where an index is not required in order to determine its location. Therefore, unnecessary loading and restoring of index registers can be avoided. See module GADSSP.

Fill-Data. These are special flags used to "patch" lapses in the telemetry stream. These lapses are caused by a variety of failures in the communication devices and related equipment. Flags are used simply to prevent processing of meaningless telemetry. For efficient use of GADS, observed input data should be edited to remove lapses and packed to reduce the main data loop to a minimum. The program symbol is FILLFL. See module GADSDA.

Fractional Change. The test for convergence in differential correction (see module GADSLS) is as follows: let u be the system parameter in question and e be the corresponding convergence criterion. Then convergence is obtained when:

$$\frac{|du|}{|u| + 100. * e} < e \qquad (1)$$

Consider the limits:

$$\lim_{u \to \infty} \frac{|du|}{|u| + 100. * e} = \left|\frac{du}{u}\right| \qquad (2)$$

and

$$\lim_{u \to 0} \frac{|du|}{|u| + 100. * e} = \frac{|du|}{100. * e} \qquad (3)$$

Formula (2) yields the fractional change in u. Formula (3) is the fractional change in u with respect to 100. * e.

Gamma. A unit of magnetic field. 1 gamma = 10^{-5} Oersted.

G.E.I. Geocentric Equatorial Inertial.

Geocentric Equatorial Inertial. A commonly used system of coordinates to which orbit and attitude can be related.

Gradient. A convenient term for partial derivatives with respect to system parameters.

Gradient Vector. In the notes of module GADSLS, the gradient vector is the gradient of the squared error function.

Ideal Prediction Function. The function giving the sensor output without accounting for any signal degrading effects caused by the sensor. See index, Reference 1.

Implicit I/O Variables. An I/O variable is implicit if it is transmitted through named common.

Implied Parameter. The first 14 fields in card G5-1, 2 may be blank and are assumed to be implied parameters. These are parameters needed in the Eulerian method of definition.

Intended System Parameters. Predicted sensor outputs and squared error functions can be requested via module GADSML for a continuous range of system parameters. As explained in the notes of module GADSLS and illustrated in Figure 7, module GADSML is invoked with several unique sets of system parameters. Each set is termed "intended" whether or not it is the one leading to least squares. The best set is not identifiable until several tests are completed.

Loop, Main Data. See Main Data Loop.

Main Data Loop. The procedure of referring to all observation times to compute the corresponding predicted functions, squared error, etc., is accomplished in the main data loop. See module GADSML.

Matrix of Coefficients. The matrix obtained in the formulation of the least squares equations of condition or normal equations. See Chapter III, Reference 1.

Method of False Position. A method of computing derivatives numerically. One method is discussed in module GADSGC. See Reference 6. Also referred to as the "regula falsi" method.

Millioersted. 10^{-3} Oersted. See Oersted.

Modified Prediction Function. See index, Reference 1.

Observable. The output of some sensor useful in attitude.

Observation. Observed data or raw data.

Observation Time. The time of an observation referred to some fixed reference or time origin.

Oersted. A unit of measure for magnetic fields.

Orbital Data. Data relating to the environment and the spacecraft ephemeris. Normally considered inputs to the attitude problem independent of the observational data.

Order of Numerical Derivative. When computing a numerical derivative using functional values obtained by adding perturbations to the argument, the number of perturbed values is called the order of the numerical derivative. (This is ually the order of the polynomial approximation to the function at the point of interest.) In GADS, however, the order is the total number of pairs of perturbed calculations computed symmetrically about the desired argument.

Outlier. Observed data is an outlier if it results in a residual exceeding the tolerance defined on card G14a-1. See Run Deck.

Pass. See Data Pass and Programmer Cycle.

Parameter. See System Parameter.

Participating Parameter or Sensor. A parameter or sensor active in a given sensor complex.

Precision. Note Accuracy.

Prediction Funcion or Operator. The function used to predict the sensor output function. See modified prediction function and index, Reference 1.

Primitive Operation. The word primitive is interpreted as non-differentiated or before differentiation.

Programmer Cycle. The hardware on board a spacecraft usually includes a programmer which, when commanded, cycles the commutators a given number of times in the proper sequence in order to obtain samples from various instruments. Usually, at the conclusion of a programmer cycle, telemetry stops and the spacecraft waits for the next command.

Quadrature. Definite integration. Certain variational equations can sometimes be expressed as quadratures. In the force-free case, the precession angle can be expressed as a quadrature. See modules GADA01 and GADSSS.

Quasi-Taylor. The normal equations employed in module GADSLS are a modification of the Taylor method of differential correction. Therefore, resulting quantities are called Quasi-Taylor.

Raw Data. See Observation.

Raw Mounting Constants. Constants (entered on cards G13a-7) which define the orientation of a sensor. They are not necessarily direction cosines but are always the quantities which the user would designate for differential correction if it is desirable to adjust the orientation of a sensor by that method. Hence, these are independent quantities. For an example, see module GADSR1.

Raw Calibration Constants. The constants (entered on cards G13a-8) which define the sensor calibration function. Like the raw mounting constants, they are not necessarily the quantities used during the processing but are those which the user can designate for differential correction. They are the independent set of constants from which the sensor's calibrating function is constructed. Such a function is discussed in modules GADC01 and GADC04.

Regula Falsi. False position. See Method of False Position.

Refining System Parameters. The process of improving the parameters by means of a correction procedure such as least-squares differential correction.

Residuals. Residuals are defined on page 19, Reference 1.

Sample Time. The time coordinate corresponding to a given observation.

Sensitive Axis. Most sensors can be thought of as having a sensitive axis. For example, the sensitive axis of a solar damage cell is the normal to the cell.

Sensor Complex. See the definition of a sensor complex in Section 3.

Sensor Function Type. The sensor output function classification. For example, a cosine sensor is discussed in module GADF01.

Sensor Library. A named COMMON area containing the permanent sensor-related information. To access a given sensor-related quantity, the sensor number or pointer is required. See module GADSSR.

Sensor Operand. The environmental phenomenon (such as solar radiation or geomagnetism) upon which the sensing instrument acts to generate its output. The sensor operand may require some special calculations. See page 82, Reference 1.

Standard Derivative. The derivatives of the ideal attitude matrix with respect to right ascension, declination and the Euler angles, or whatever angles are employed to define the attitude. (Hence, these standard derivatives are analogous to the derivatives of the observations range, azimuth,... with respect to orbital position.) They are standard because of the simplicity of generation and application. They do not involve the chain rule of differentiation and often suffice for the attitude determination problem.

State Variable. See page 36, Reference 1. Refer to modern discussions on orbit determination and optimization techniques.

System Parameters. Constants which define a system. They may be initial conditions to the equations of motion, calibration constants, mounting constants, phase angles, frequencies, etc. Although parameters are constants, they may be refined by differential correction.

Taylor Method. See Reference 1.

Taylor Method, Quasi. See notes, module GADSLS. See also Quasi-Taylor.

Telemetry Counts. The information telemetered from an artificial satellite to a ground station is usually scaled. This means that it has been transformed to telemetry units or counts from engineering units.

Time Origin. The reference time to which all observation sample times are referenced during a pass through GADS. It is the double precision time of year, TFIXED. See module GADSLS and

Time Base. Time Origin.

Variational Equations. Let X represent the state of a system. (For example, in orbit determination, $X = \left\{ x, y, z, \ x, y, z \right\}$.) This so-called state variable satisfies the differential equations of motion, namely:

$$\dot{X} = F \ (X, \ U, \ t),$$

where F is a vector function of X, the time t, and U (the array of system parameters. That is, $U = \left\{ u_1, u_2, \ldots u_n \right\}$.

Consider the expression:

$$\frac{\partial \dot{X}}{\partial u_k} = \frac{\partial F}{\partial X} \cdot \frac{\partial X}{\partial u_k} + \frac{\partial F}{\partial u_k} .$$

This expression is an example of a variational equation. Variational equations can be integrated to yield the partial derivatives needed in differential correction.

Weighting Factors and Functions. Refer to pages 13 and 14, Reference 1. Weighting factors are applied automatically in the GADS system. This factor is entered on card G13-9 and can be subordinated to a weighting function if requested on cards G14-2, 3. Weighting functions are computed in modules of the type GADW01.

SECTION 6

NEW APPLICATIONS

6.1 INTRODUCTION

An important objective of GADS is to provide a convenient tool for handling new and unexpected attitude determination problems. To meet this objective, GADS was designed so that the executive superstructure is concerned only with the mechanization of least squares differential correction. Conversely, worker modules are concerned only with the specifics of a given application. To apply GADS to a novel problem, the user need only be concerned with the worker modules. Furthermore, he should not take it as a foregone conclusion that he will need to add worker modules at the drop of a new sensor. Most sensors are based on a few simple ideas and a new sensor does not necessarily imply a new type of output function. Note, for example, that the same module, GADF01, serves most of the sensors involved in the EPE-D and ISIS-A applications. Note also that module GADF02, which handles the problem of shadowing, is a minor modification of GADF01. When the user determines the need for a new worker module, he should consider whether a minor modification of an existing module will be acceptable. The following paragraphs have been included to aid the user in modifying the GADS capabilities.

6.2 DIVISION OF WORK

To simplify the problem of identifying the desired worker modules, they are discussed below from the point of view of their purpose. In the following discussion the characters XX will mean 01, 02, 03, etc. The discussion begins at the lowest working level.

6.2.1 MODULES GADAXX AND GADDXX

These modules calculate auxiliary functions. A typical auxiliary function is the state vector. That is, these modules operate on the system parameters to obtain the intermediate results which define the system state (e.g., the attitude) as a function of time. These modules are dependent on the parametrization since they assume a definite set of input arguments. In so far as the parameters of the motion are concerned, GADS is, at this writing, prepared only for problems defined in terms of Euler angles. See paragraph 6.3.2.

6.2.2 MODULE GADSAL

Using the state vector, this module calculates the attitude matrix and its simple derivatives. The existing module GADSAL will require modification when handling problems not defined in terms of Euler angles.

6.2.3 MODULE GADSSO

This module prepares the sensor operand. The user should insure that this module calculates the arguments derived from the environment; i.e., from independent information. A typical sensor operand is the geomagnetic field in the vicinity of the spacecraft, for example. These results are used by the modules discussed below in paragraph 6.2.4.

6.2.4 MODULES GADFXX AND GADGXX

These modules combine the results from GADSSO and GADSAL to obtain f, the ideal sensor prediction function and its derivatives with respect to the system parameters. Ideally, module GADFXX should be programmed to compute f and all of its derivatives. Module GADGXX can also be used to compute peculiar derivatives. Note also that derivatives with respect to calibration constants are handled by modules GADCXX.

6.2.5 MODULES GADCXX.

These modules transform the ideal sensor output computed in GADFXX to account for non-ideal effects, such as non-linearity, bias, etc. They should also be able to compute the various derivatives that may be needed.

6.3 EXPANDING MODULE GADSIN

Module GADSIN interprets the simple English-like statements in the GADS Run Deck. When the user adds new worker modules or makes program improvements, he may avoid modifying GADSIN if the full capabilities of the Run Deck are used; i.e., if all module numbers are stated explicitly. An alternative method is to incorporate new control codes in GADSIN so that it can perform more of the clerical work.

Expanding the scope of GADSIN is simplified since no calculations are involved. Because this module is a translator, its main function is to perform a series of table-look-ups. Therefore, the problem of expanding the capabilities is that of expanding the tables (increasing the vocabulatory). The purpose of this discussion is to describe the key tables (arrays) which the user may wish to expand. A table may have two parts: a table of comparands and the corresponding octal codes or GADS synonyms. The comparands are the English-like codes which constitute the vocabulatory and which the user wishes to be recognized by GADS. Sometimes the synonym is simply the work number or position so that the second part of the table is obviated.

6.3.1 SENSOR-RELATED ARRAYS

First, consider those arrays related to sensors: output and calibration functions, operands, and mounting constants.

Table of Resident Comparands	Contents (Hollerith)	Table of Synonyms	Contents (Octal)	Explanation
SCODE 1		ICODE 1	NNNN	These tables identify the sensor operand and the corresponding internal GADS pointer code, respectively. During processing, JSCODE will be set to NNNN. See modules GADFXX and GADSSO.
	SOLAR		0001	Solar line-of-sight.
	MAGNET		0002	Magnetic field vector.
Suggested for future programming.				
	HORIZN		0003	Horizon for I.R. scanners.
	STELLR		0004	Star line-of-sight.
	PLASMA		0005	Plasma relative velocity direction.
	GROUND		0006	Ground station line-of-sight for optical instruments.
SCODE 2		ICODE 2	FF GG	These tables identify the sensor prediction function and the GADS modules used to perform the calculations, respectively. The octal characters FFGG are high order and identify the modules GADFXX and GADGXX, respectively.
	COSINE		0101	For cosine sensors. See module GADF01.
	TUNED		0202	For tuned oscillator output function (not programmed). See page 84, Reference 1. See also SCODE 3, below.

Table of Resident Comparands	Contents (Hollerith)	Table of Synonyms	Contents (Octal)	Explanation
SCODE 3		(not needed)	ĴĴ	This table contains the types of calibration functions. ĴĴ selects module GADCĴĴ.
	POLYN.		01	Polynomial calibration. See Notes, module GADC01.
	FURİER		02	Fourier calibration. (At this writing, GADC02 is not programmed.)
	OSCİL.		03	Oscillator magnetometer output. See Reference 1, page 84. (Module GADC03 is not programmed.)
	POLYN1		04	Polynomial calibration. See Notes, module GADC04.
SCODE 4		(not needed)	ĴĴ	This table contains the types of sensor mounting. JJ selects modules of the type GADSR1, 2,... See module GADK01.
	FİXEDV		01	Fixed vector type mounting. See Figure 16 and module GADSR1.
	AXİS		02	This code could be used to define a wheel. (Not programmed.)

6.3.2 PARAMETRIZATION

Module GADSIN can be modified to handle other parametrizations besides the Eulerian method. The important arrays are PCODES (32, NPDEFM) and NPCODE (6, NPDEFM). The variable NPDEFM is a FORTRAN PARAMETER. Its purpose is to determine the dimensionality of these arrays and to limit certain "DO" loops within the module GADSIN. The value of NPDEFM (number of parametrization definitions, maximum) should equal the total number of parametrizations which module GADSIN is programmed to interpret.

The key arrays are described in the following table. It should become evident, though perhaps not too quickly, how the user can incorporate his own parameters. First, the parameter names should be loaded into PCODES (I, 2), I=1, 2, 3,... A third parameter set could be loaded into PCODES (I, 3), etc. The nonstandard derivative codes should be entered into the corresponding cells of NPCODE, noting that for standard derivatives KK = 00. Non-standard derivatives are computed by modules of the type GADGXX; XX = 31, $32_8, \ldots 50_8$. When differentiation with respect to a parameter is to be computed only by the method of false position, KK should be greater than 30_8 in order to avoid a diagnostic. This statement applies to all parameters for which derivative functions are not programmed. For an example, see the parameters discussed in the following table beginning with AAA.

Table of Resident Comparands	Contents (Hollerith)	Table of Synonyms	Contents (Octal)	Explanation
PCODES		NPCODE	KK	PCODES ($\dot{I}$, 1) contains the hollerith names of the Euler angles, their rates, moments of inertia, and other key parameters. NPCODE ($\dot{I}$, 1) is used to give the selection codes for the derivatives which are nonstandard. Hence the code KK selects a module of the type GADGXX. Standard derivatives are computed in modules GADFXX. Codes 31_8 are used to "trick" GADSIN into accepting a "legal" code even though no module GADG25 exists.
	ALPHA		00	Standard derivative, no code needed.
	DELTA		00	Standard derivative, no code needed.
	PHİO		00	Standard derivative, no code needed.
	THETAO		32	See module GADG26.
	PSİO		00	Standard.
	PHİDOT		33	See module GADG27.
	PSİDOT		50	See module GADG40.
	AAA		31	Derivatives w.r. to the ensuing parameters are done numerically which means these codes are ignored. See card G14-c.
	BBB		31	
	CCC		31	
	etc.		etc.	

SECTION 7

REFERENCES

1. Crocker, John P.: "Differential Correction Methods in Spacecraft Attitude Determination", NASA contractor report WD-9758-21-2, June 17, 1968.

2. Marquardt, Donald W.: "An Algorithm for Least-Squares Estimation of Nonlinear Parameters", J.S.I.A.M. 11 (2), 431, June 1963.

3. Abramowitz, M. and Stegun, I.A.: "Handbook of Mathematical Functions", N.B.S. Applied Mathematics Series 55, 1965.

4. Ames, J. S. and Murnaghan, F.D.: "Theoretical Mechanics", Dover, 1953.

5. Leimanis, E.: "The General Problem of the Motion of Coupled Rigid Bodies About a Fixed Point", Springer-Verlag, 1965.

6. Henrici, P.: "Elements of Numerical Analysis", John Wiley & Sons, 1964.

7. Glauz, R.D.: "Partial Step Integration", C. of A.C.M., 7 (9), July 1966.

8. "UNIVAC 1108 Central Computer General Manual", UNIVAC publication UP2463, review 2.

9. Cain, J. and Hendricks, S.: "Computation of the Main Geomagnetic Field from Spherical Harmonic Expansions", NASA X-611-64-316, October 1964.

10. "SC 4020 Microfilm Recorder User's Manual": Computer Sciences Corporation Report, NAS5-9758-39.

11. MacLaren, M.D. and Marsaglia, G.: "Uniform Random Number Generators", J.A.C.M. 12 (1), January 1965, pp. 83-89.

12. Hammersley, J. M. and Handscomb, D.C.: "Monte-Carlo Methods", Methuen and Co., Ltd., and John Wiley & Sons, 1964.

13. "Explanatory Supplement to the Astronomical Ephemeris and American Ephemeris and Nautical Almanac", H. M. Nautical Almanac Office, London. Obtainable from British Information Services, 45 Rockefeller Plaza, N. Y., N. Y., 10020.

14. "American Ephemeris and Nautical Almanac", U. S. Government Printing Office.

15. Jensen, J., Townsend, G., et al: "Design Guide to Orbital Flight", McGraw-Hill, 1962.

16. Quinichett, R. L.: "Statistical Confidence Measures of Least-Squares Parameters", NASA report WD-9758-159-1, February 19, 1969.

17. Smart, W. M.: "A Textbook on Spherical Astronomy", Cambridge University Press, 1965.

18. Allen, R., and Groover, C.: "Special Attitude and Orbit Extract Program for the Energetic Particle Explorer Satellite (EPE-D), NASA X-565-68-499, December 1968.

19. Miller, A. S., and Taylor, W. M. P.: "Generalized Mass Storage I/O Handler" (IOHAND), WD-9758-88-3.

20. Barry, P. F., et al: "EPE-D Supplementary Programming", August 1, 1965, IBM Corporation.

21. Lewis, Goodman, Miller: "A Pseudo Random Number Generator for the IBM 360", I. B. M. Systems Journal 8, (2), 1969, page 147.

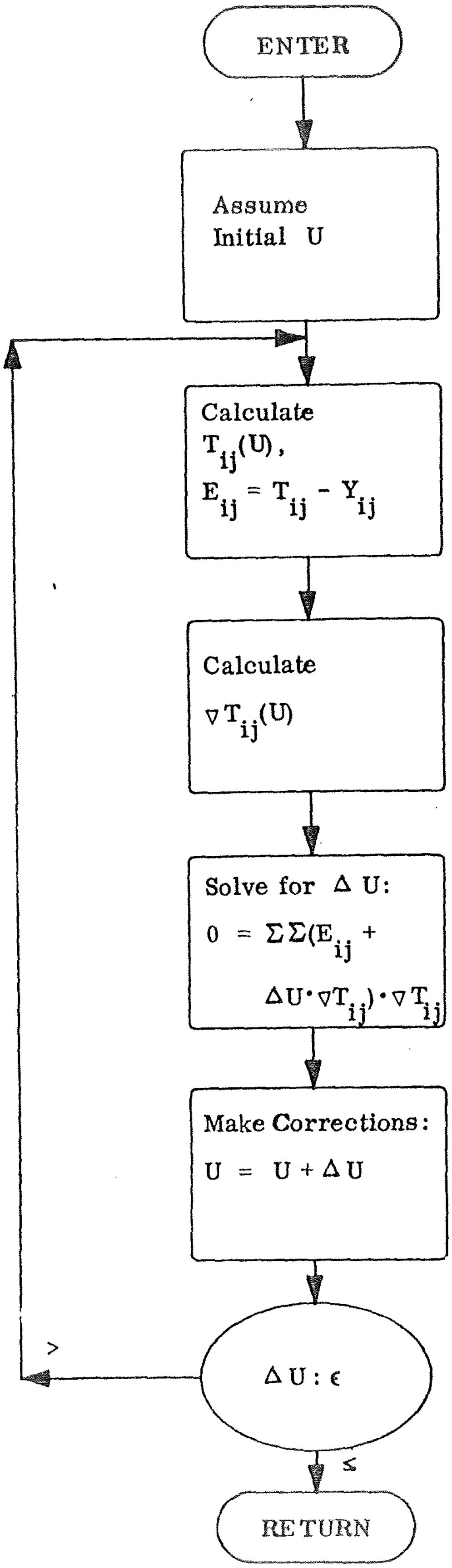


Figure 1. Newton Differential Correction Procedure In Several Variables for Vector Functions

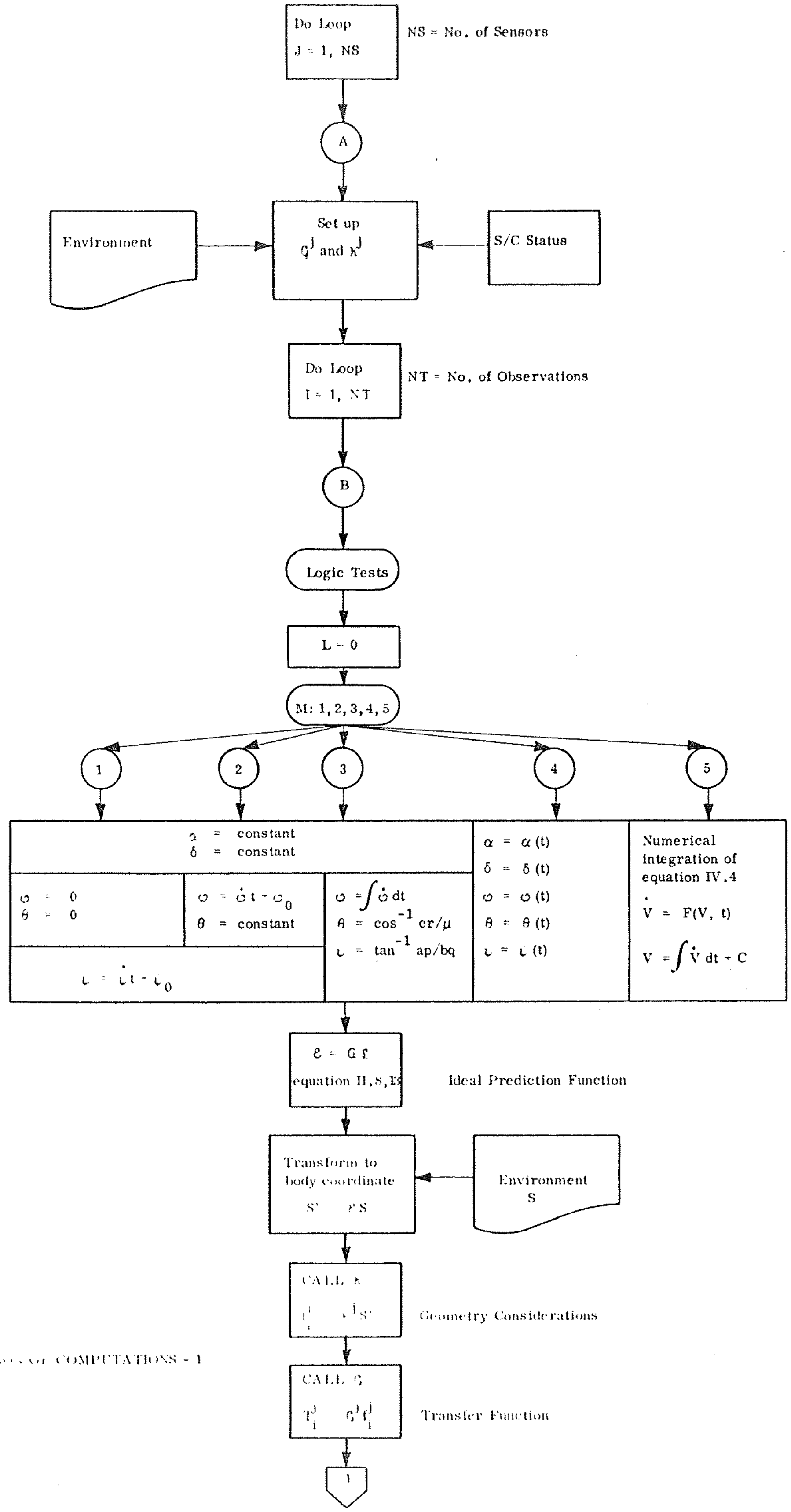
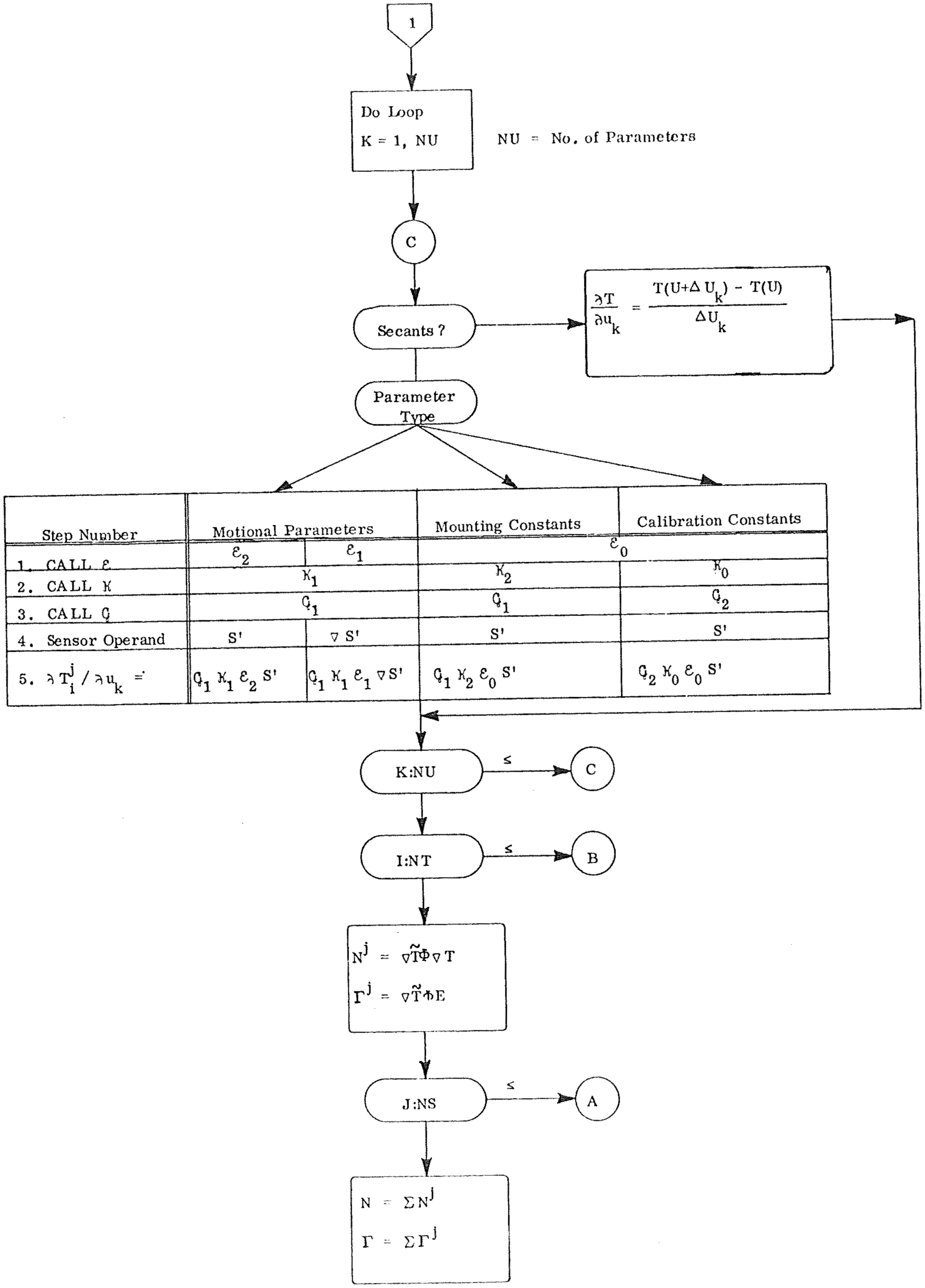


Figure 2. Mechanization of Computations - 1



Step Number	Motional Parameters		Mounting Constants	Calibration Constants
1. CALL ε	ε_2	ε_1	ε_0	
2. CALL κ	κ_1		κ_2	κ_0
3. CALL G	G_1		G_1	G_2
4. Sensor Operand	S'	∇ S'	S'	S'
5. $\partial T_i^j / \partial u_k$ =	$G_1 \kappa_1 \varepsilon_2 S'$	$G_1 \kappa_1 \varepsilon_1 \nabla S'$	$G_1 \kappa_2 \varepsilon_0 S'$	$G_2 \kappa_0 \varepsilon_0 S'$

Figure 3a. Mechanization of Computations - 2

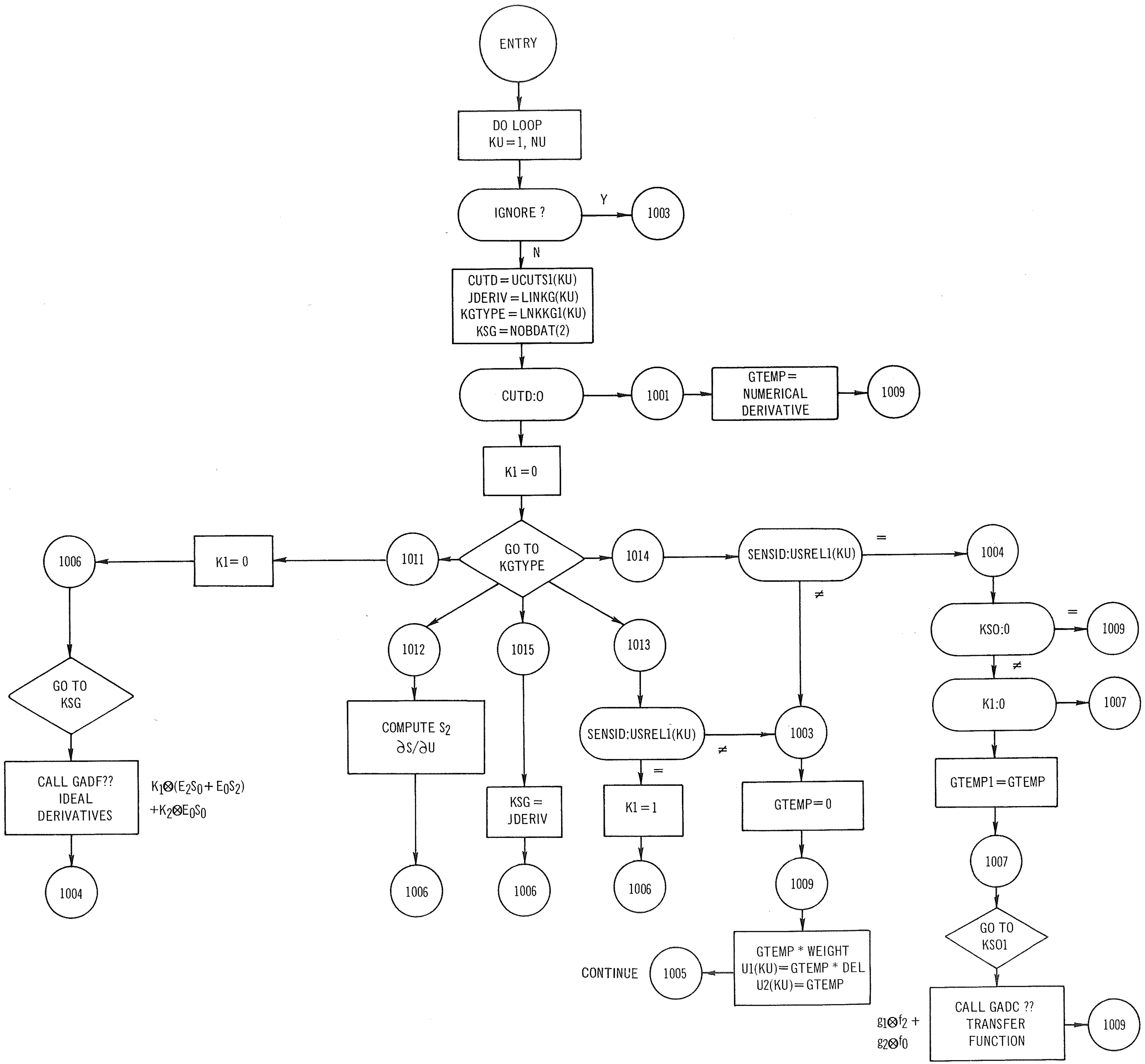


Figure 3b. Computation of the Derivatives Module GADSGC

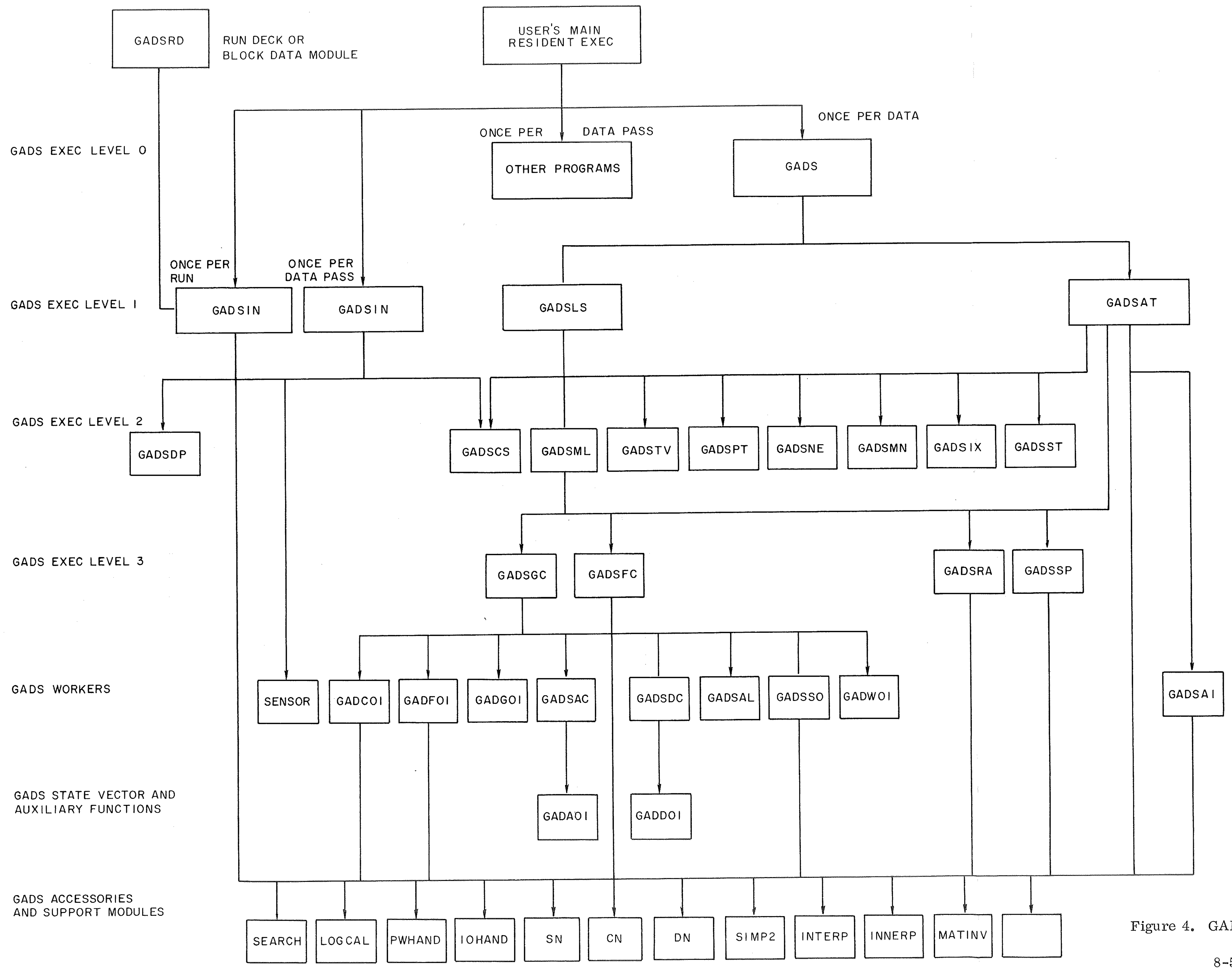


Figure 4. GADS Hierarchy

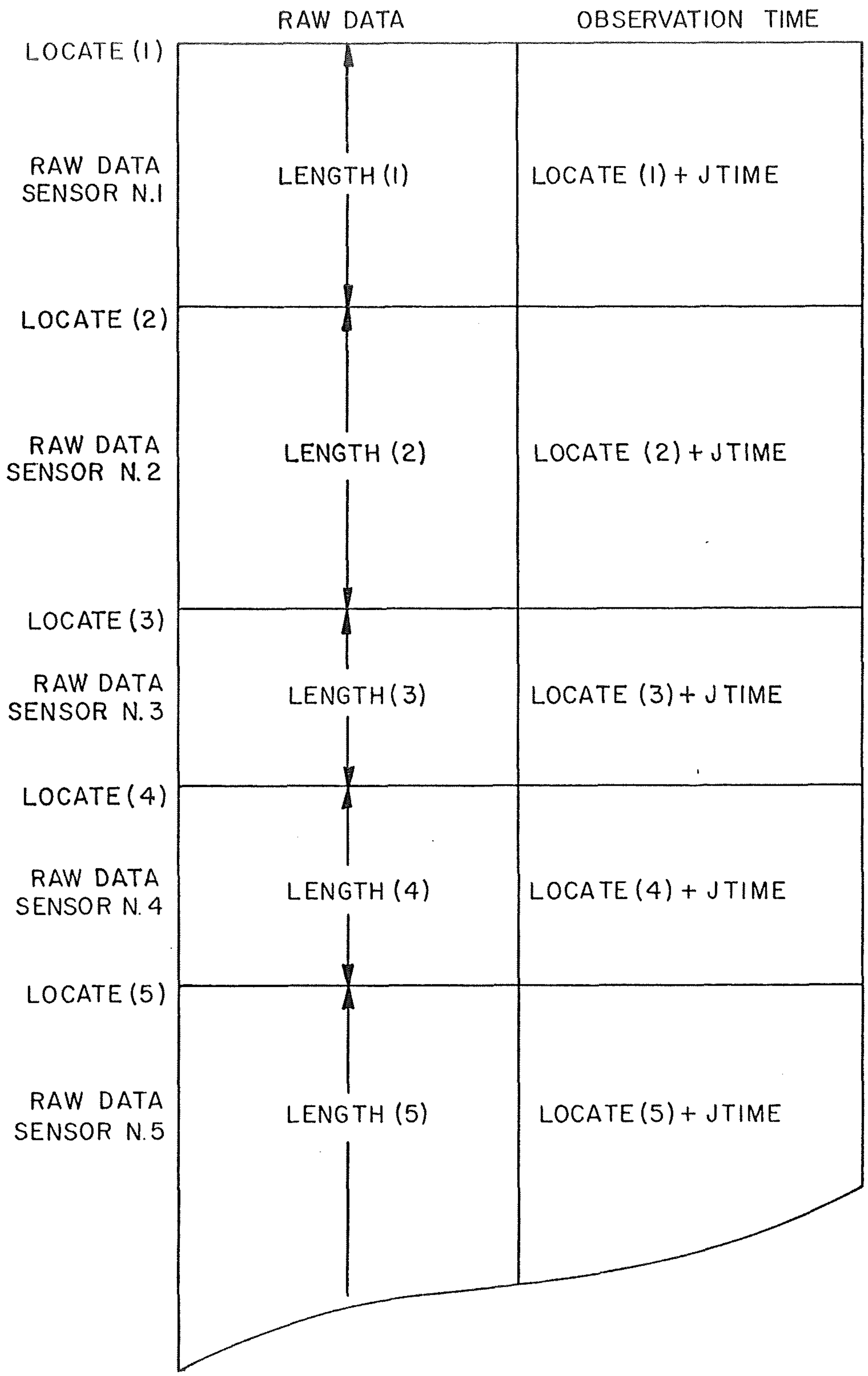


Figure 5. Location of Raw Data and Sample Times in Array COMM. See Usage and Modules GADSLS, GADS/ISISA.

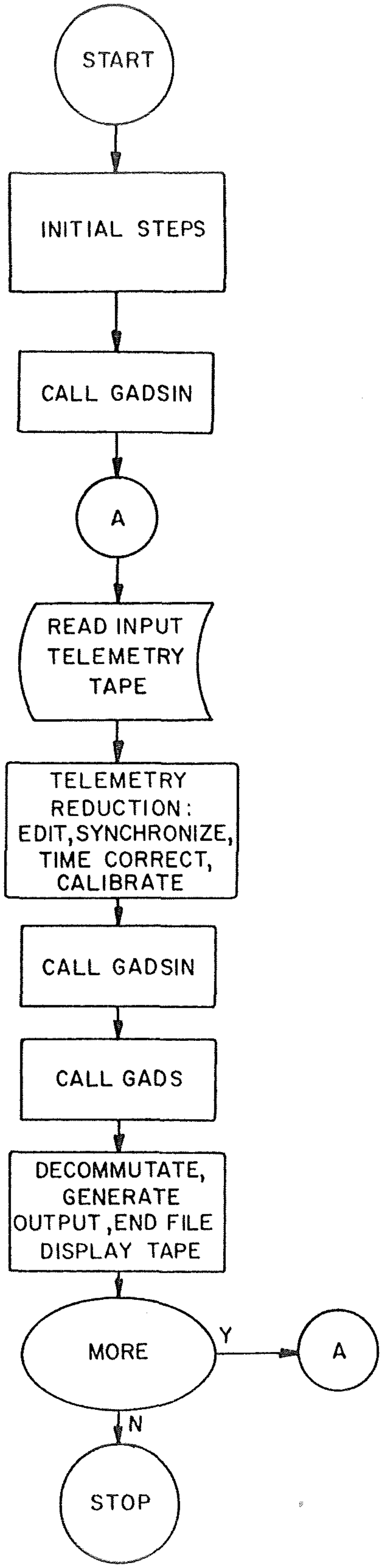
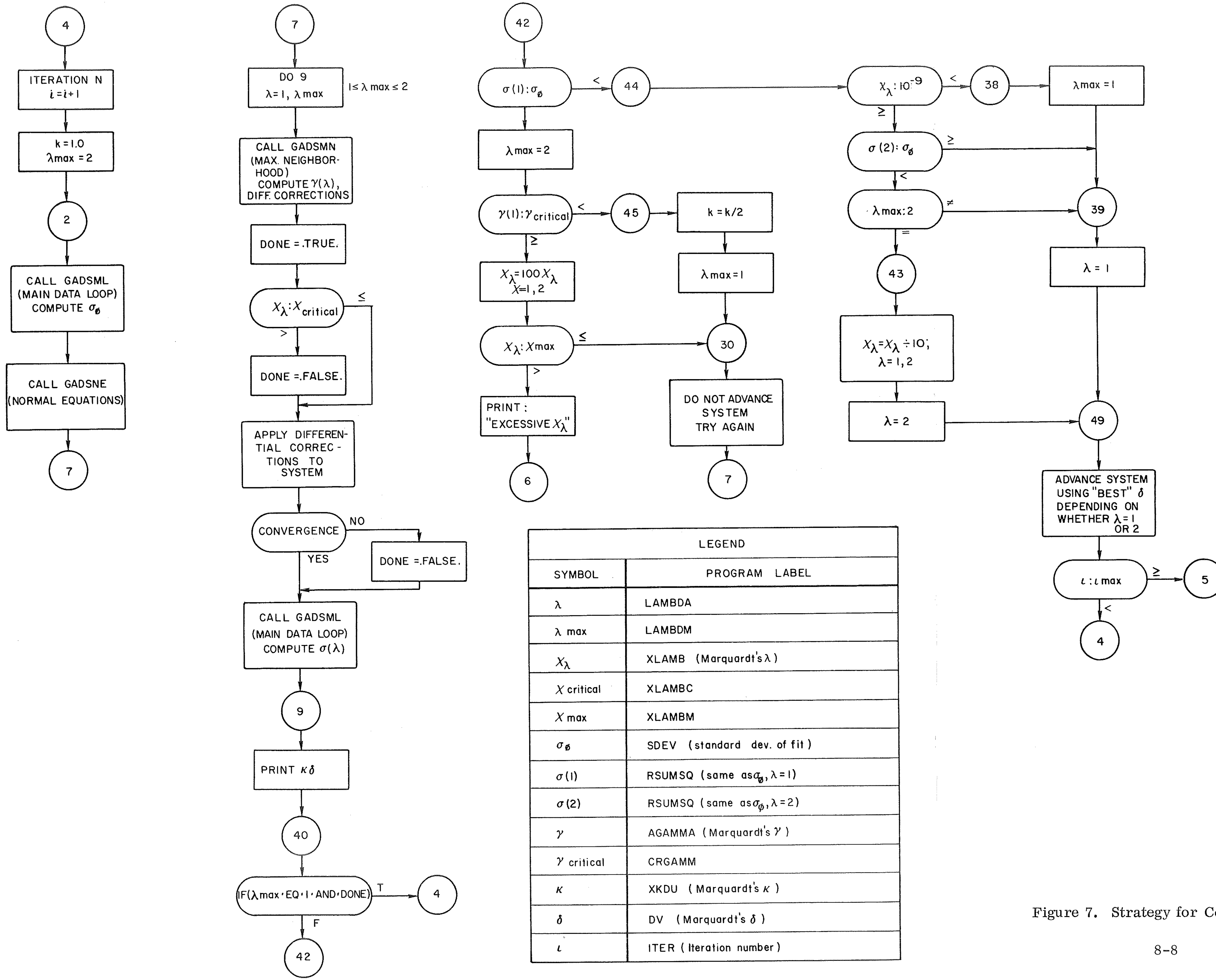


Figure 6. High Level Processing Flow.
See Module GADS/ISISA.



LEGEND	
SYMBOL	PROGRAM LABEL
λ	LAMBDA
λ max	LAMBDM
X_λ	XLAMB (Marquardt's λ)
X critical	XLAMBC
X max	XLAMBM
σ_ϕ	SDEV (standard dev. of fit)
$\sigma(1)$	RSUMSQ (same as σ_ϕ, $\lambda = 1$)
$\sigma(2)$	RSUMSQ (same as σ_ϕ, $\lambda = 2$)
γ	AGAMMA (Marquardt's γ)
γ critical	CRGAMM
κ	XKDU (Marquardt's κ)
δ	DV (Marquardt's δ)
ι	ITER (Iteration number)

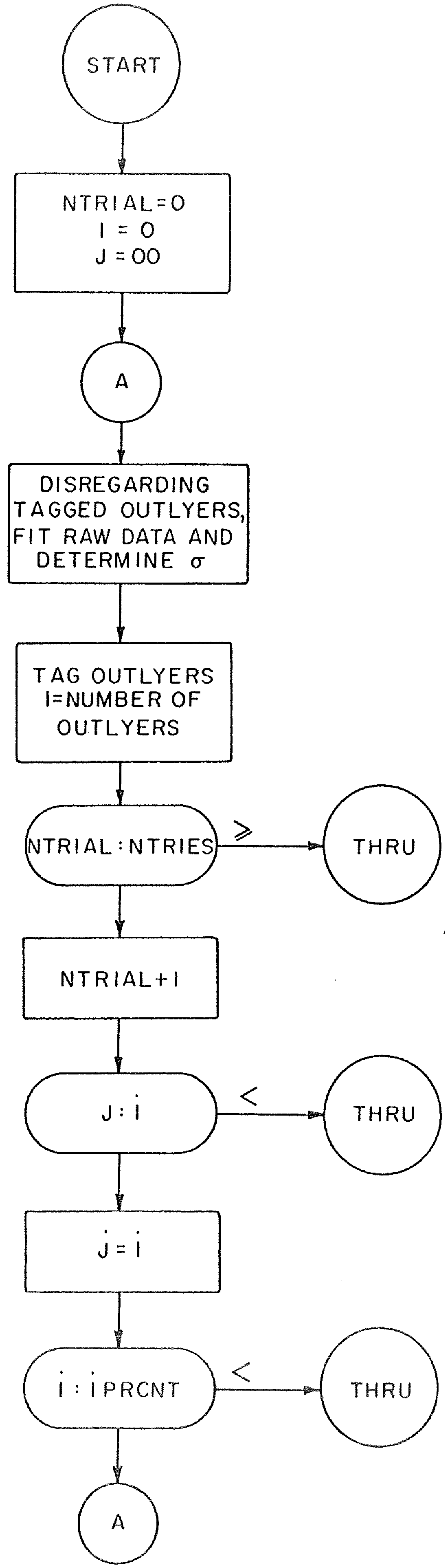
Figure 7. Strategy for Convergence

MODULE	ACTION	COMMENTS
GADSLS: GADSIX	USC → U, U1, U2	INITIALIZATION ONLY
	PARAMS → VNAMES	
	UCUTS → UCUTS1	
	EPS → EPS1	
	USREL → USREL1	
U → UTEMP		
GADSML	USING UTEMP, COMPUTE RESIDUALS AND σ_0	
GADSNE	NORMALIZE COEFFICIENTS	
GADSMN	COMPUTE MARQUARDT'S δ, NAMELY DV, AND CORRECT:	
	U1 = U1 − DV ; STEEPEST DESCENT,	
	U2 = U2 − DV ; TAYLOR METHOD	
U1 → UTEMP ; STEEPEST DESCENT,		
U2 → UTEMP ; TAYLOR METHOD		
GADSML	USING UTEMP AGAIN, COMPUTE RESIDUALS AND σ_0	

DO ONE OF THE FOLLOWING:

1. U → UTEMP; RESTORE SYSTEM
2. U1 → UTEMP; ADVANCE USING STEEPEST DESCENT
3. U2 → UTEMP; ADVANCE USING QUASI-TAYLOR VECTOR

Figure 8. Flow of System Parameters During Differential Correction


START
NTRIAL=0
I = 0
J = 00
A
DISREGARDING TAGGED OUTLYERS, FIT RAW DATA AND DETERMINE σ
TAG OUTLYERS
I=NUMBER OF OUTLYERS
NTRIAL : NTRIES
≥
THRU
NTRIAL+1
J : i
<
THRU
j = i
i : iPRCNT
<
THRU
A

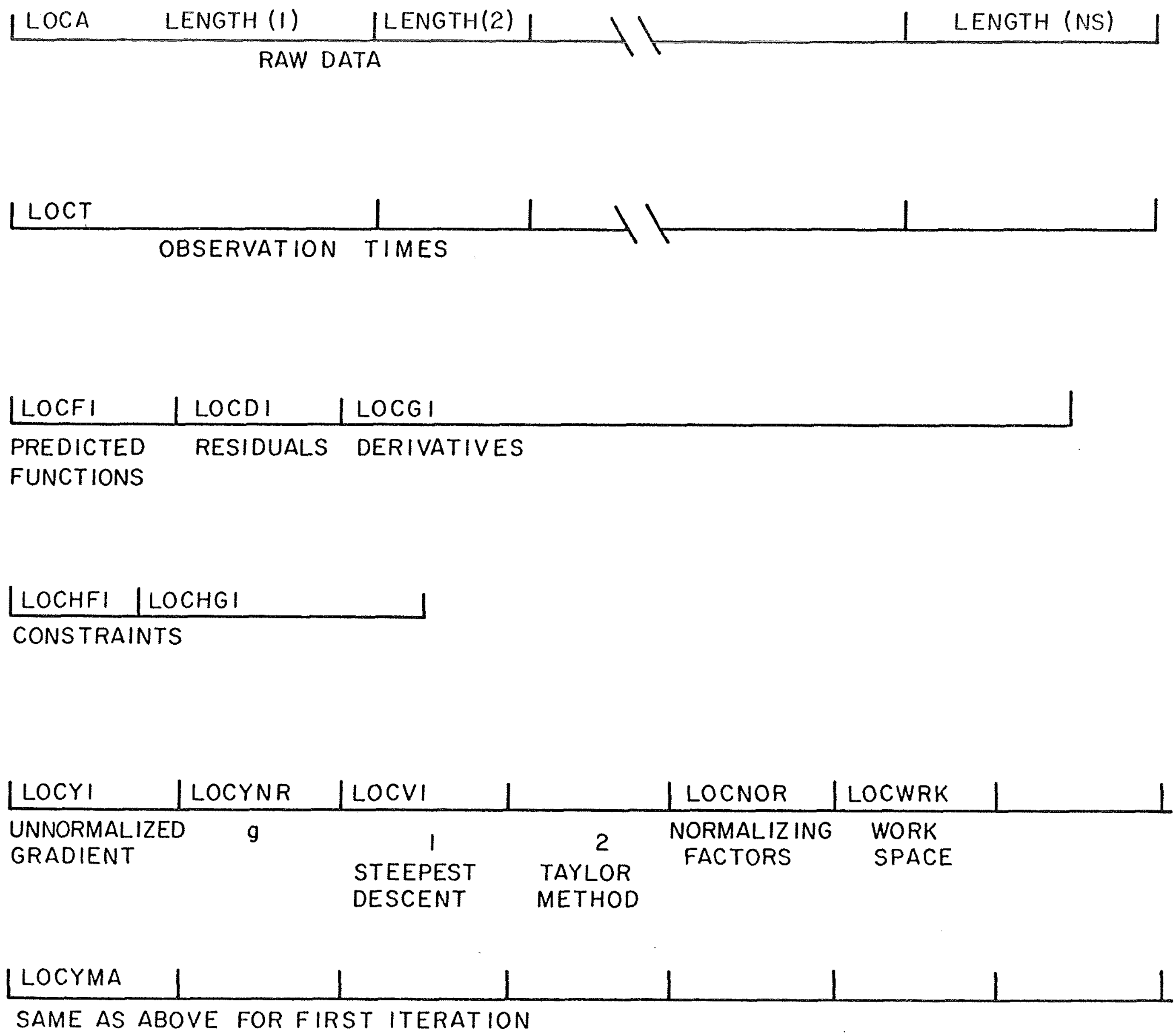


Figure 10. Layout of Array COMM
(Sheet 1 of 2)

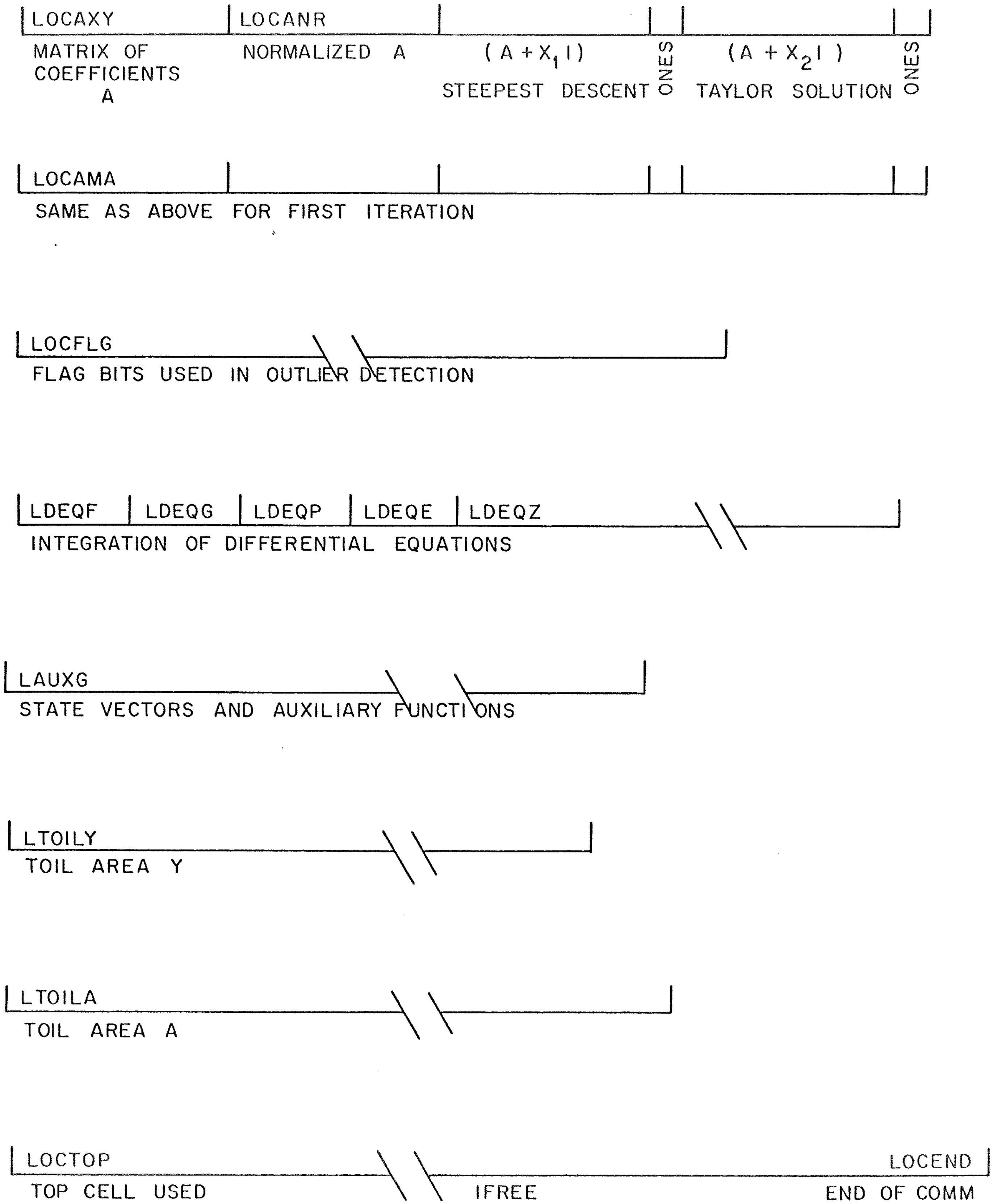


Figure 10. Layout of Array COMM
(Sheet 2 of 2)

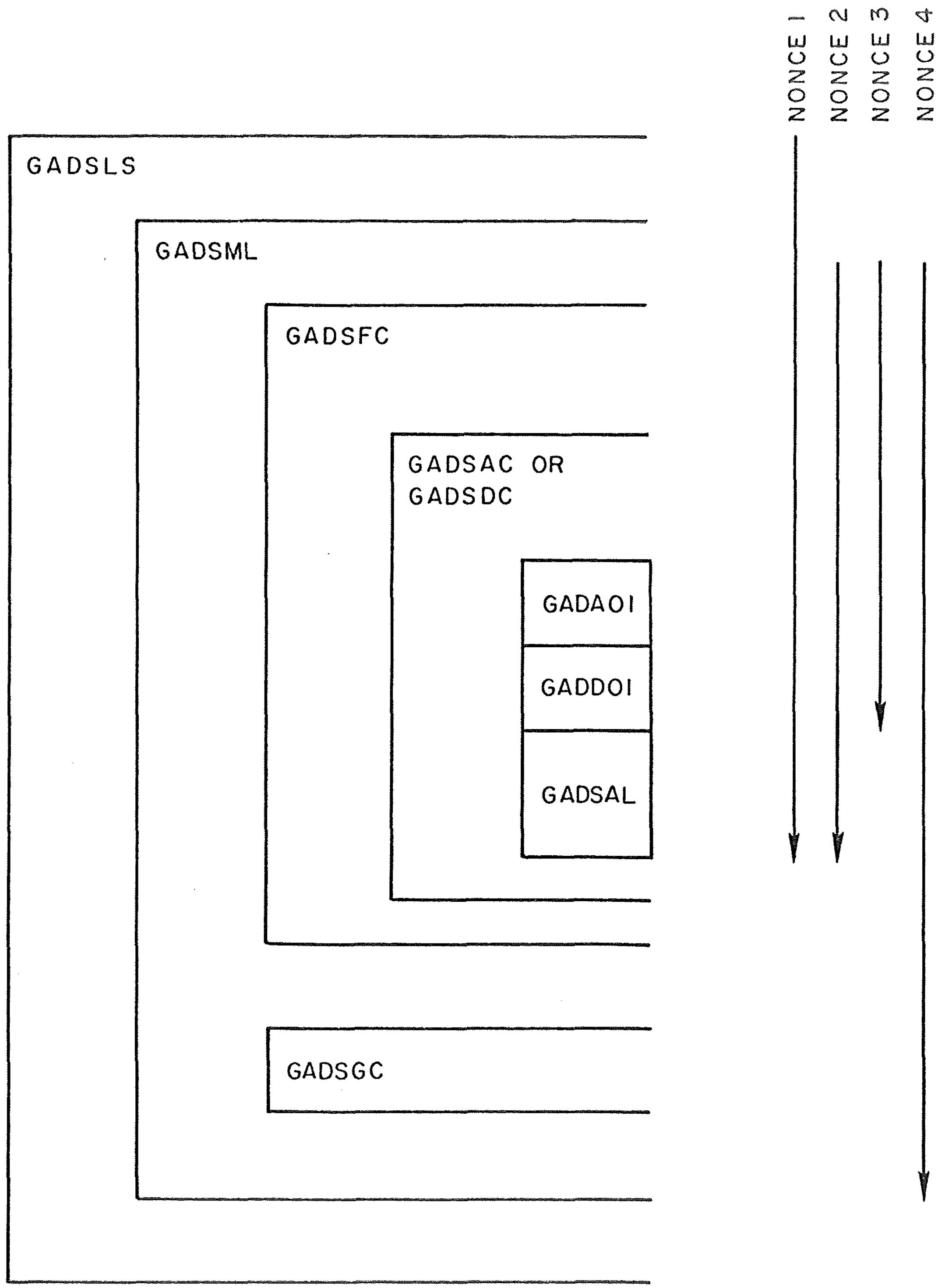


Figure 11. Ranges for Variables NONCE

WORD \ ARRAY	LINKF		
1	000	000	A
2	000	000	B
3	000	000	C
4	000	000	D
5	000	000	E
6	000	000	F
7	000	000	G
8	000	000	H
9	000	000	000
10			

WORD \ ARRAY	NCF		
1	A	B	C
2	D	E	F
3	G	H	000
4			
5			

```
   DO 1  i = 1, 8
1. CALL STHRDS (LINKF, NCF, i, 3 )
   DO 2 i = 1, 8
2. CALL LTHRDS (NCF, LINKF, i, 3 )
```

Figure 12. Illustration of PWHAND Loading and Unloading Linkages. See Module PWHAND.

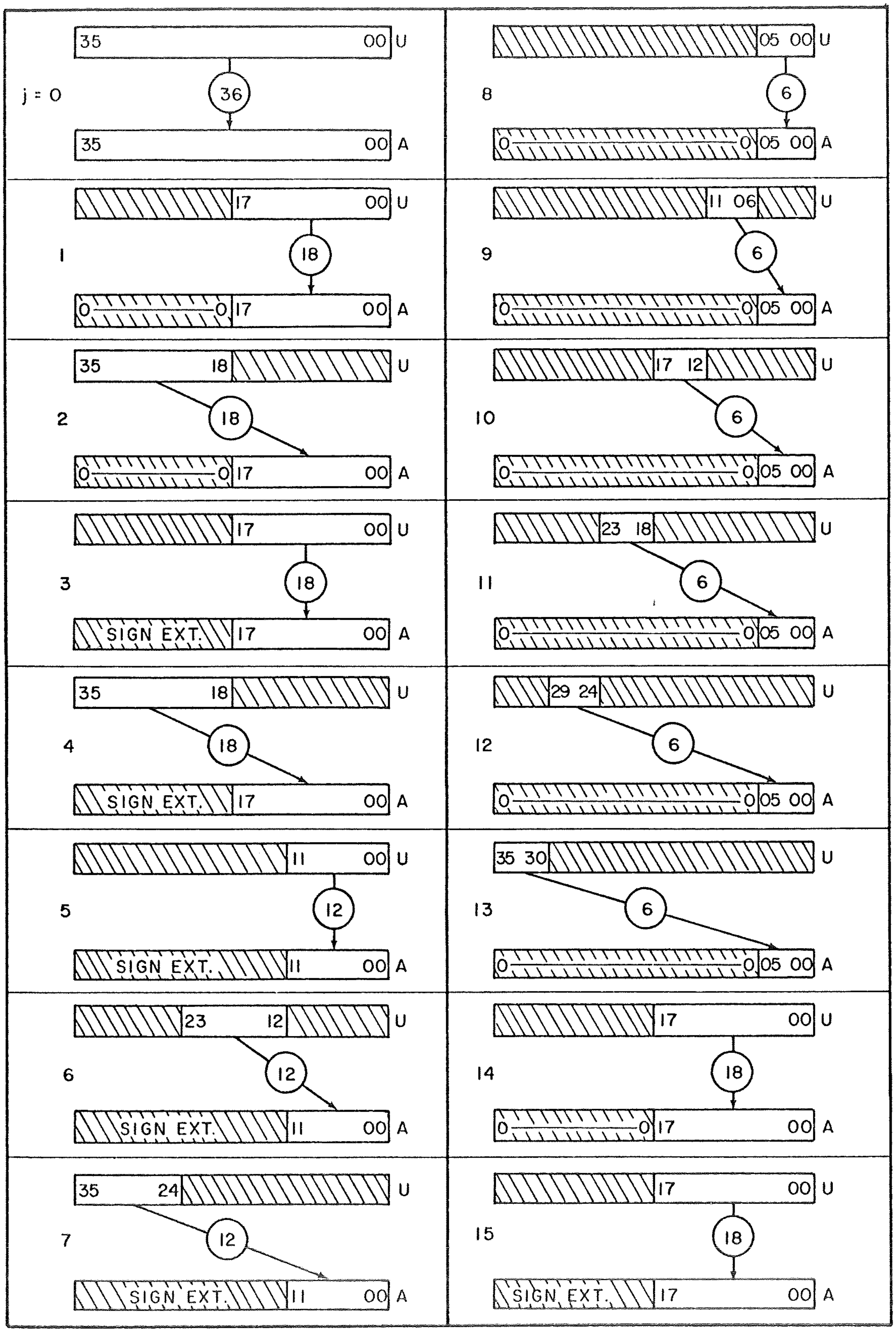


Figure 13-a. Partial Word Capability of UNIVAC 1108.
See Module PWHAND and LOGCAL.

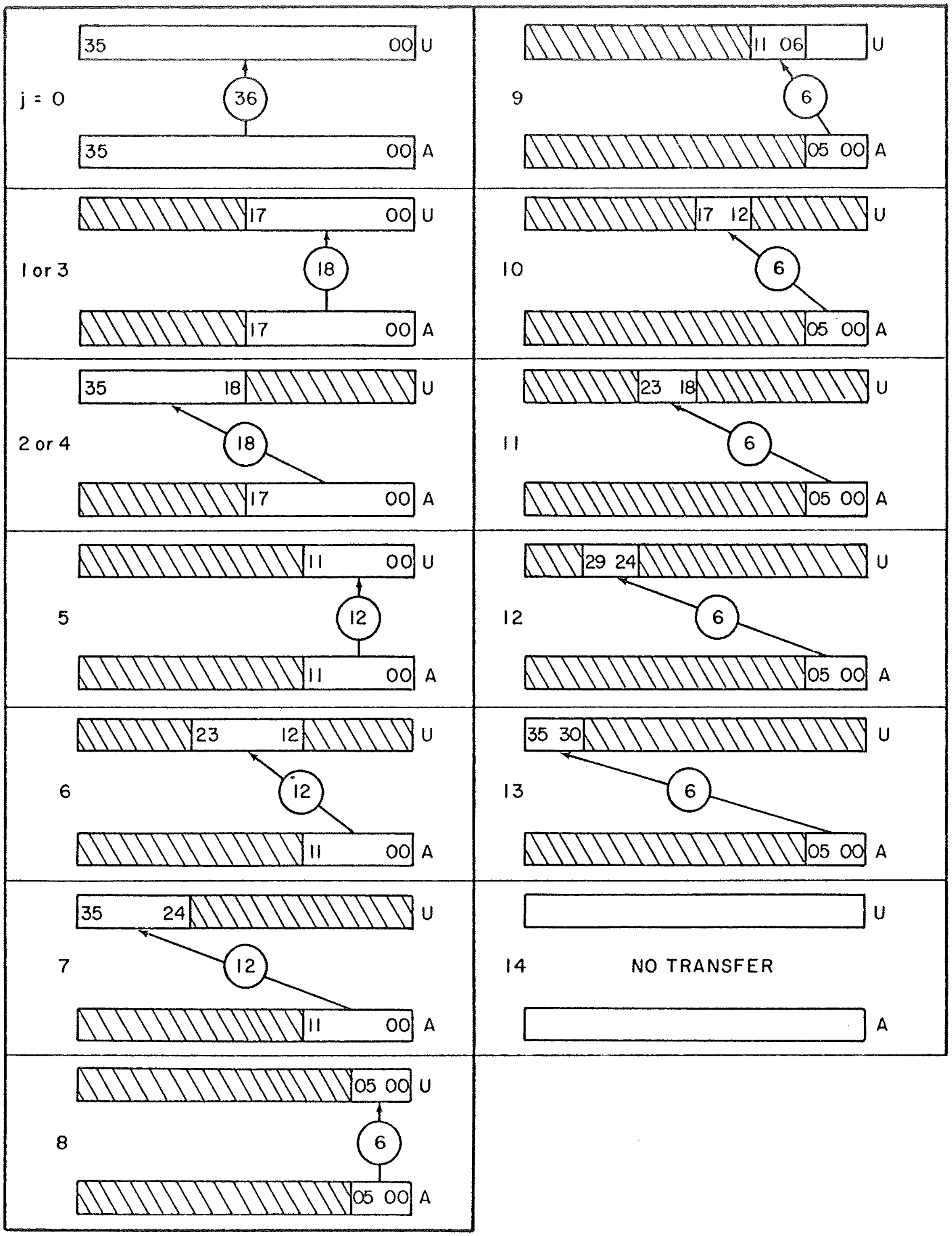


Figure 13-b. Partial Word Capability of UNIVAC 1108.
See Module PWHAND and LOGCAL.

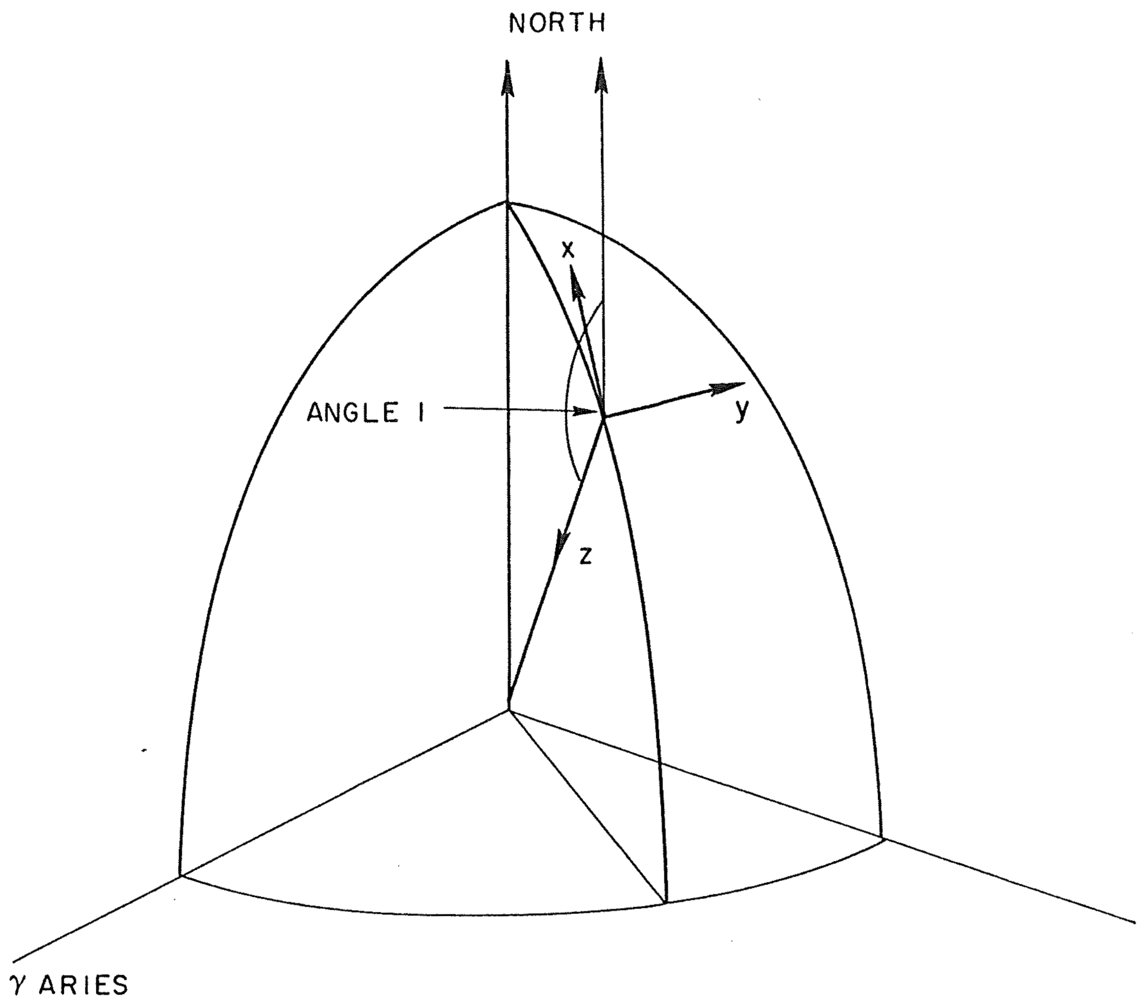


Figure 14. The Vernal Equinox System and the Local North-East-Down System. See Module LOCAL1.

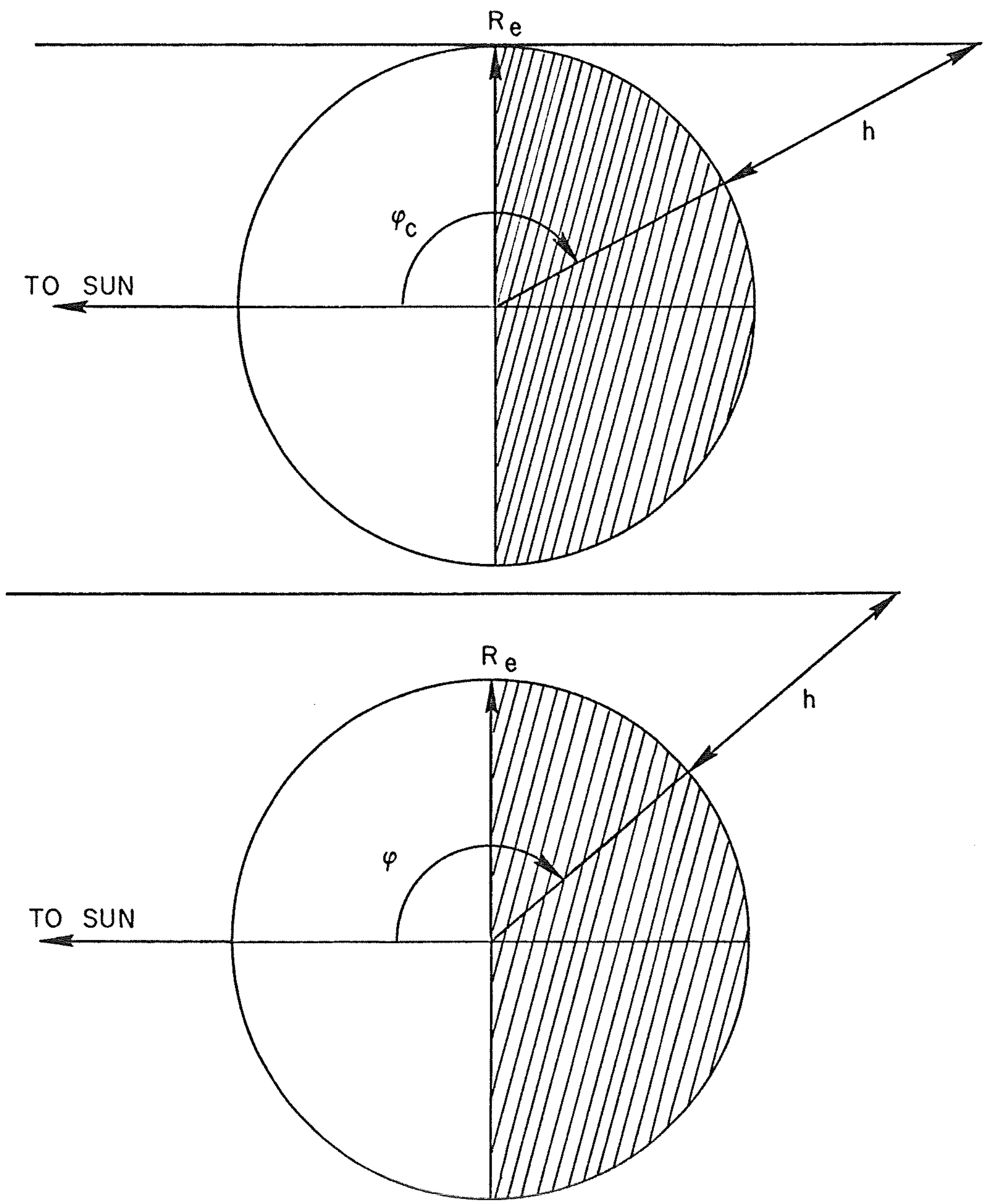


Figure 15. Calculation of Shadow Angle.
See Module GADSSO/EPED.

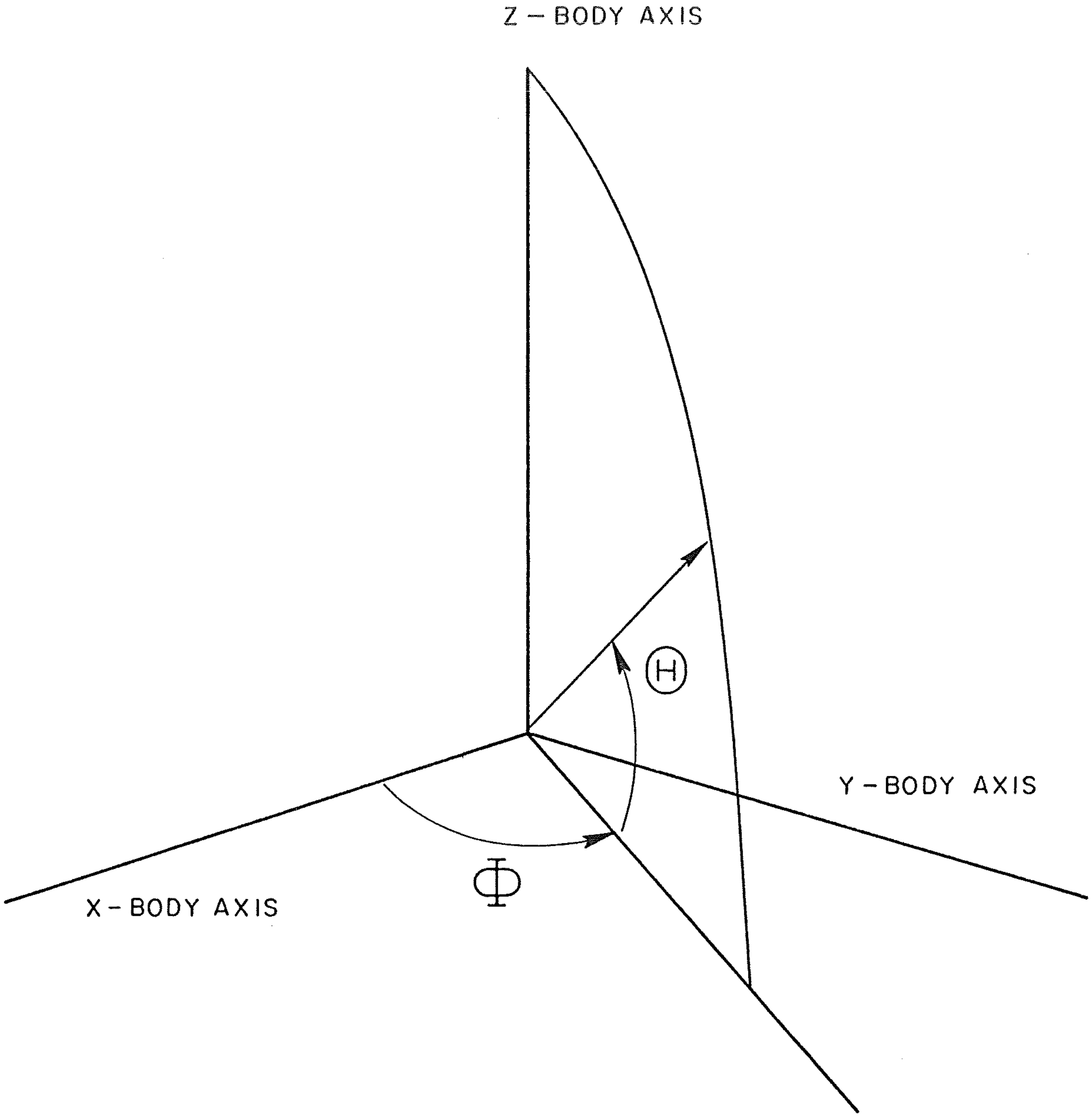


Figure 16. Mounting or Geometric Constants for a Fixed Cartesian Line-of-Sight Sensor. See Modules GADSR1, SENSOR.

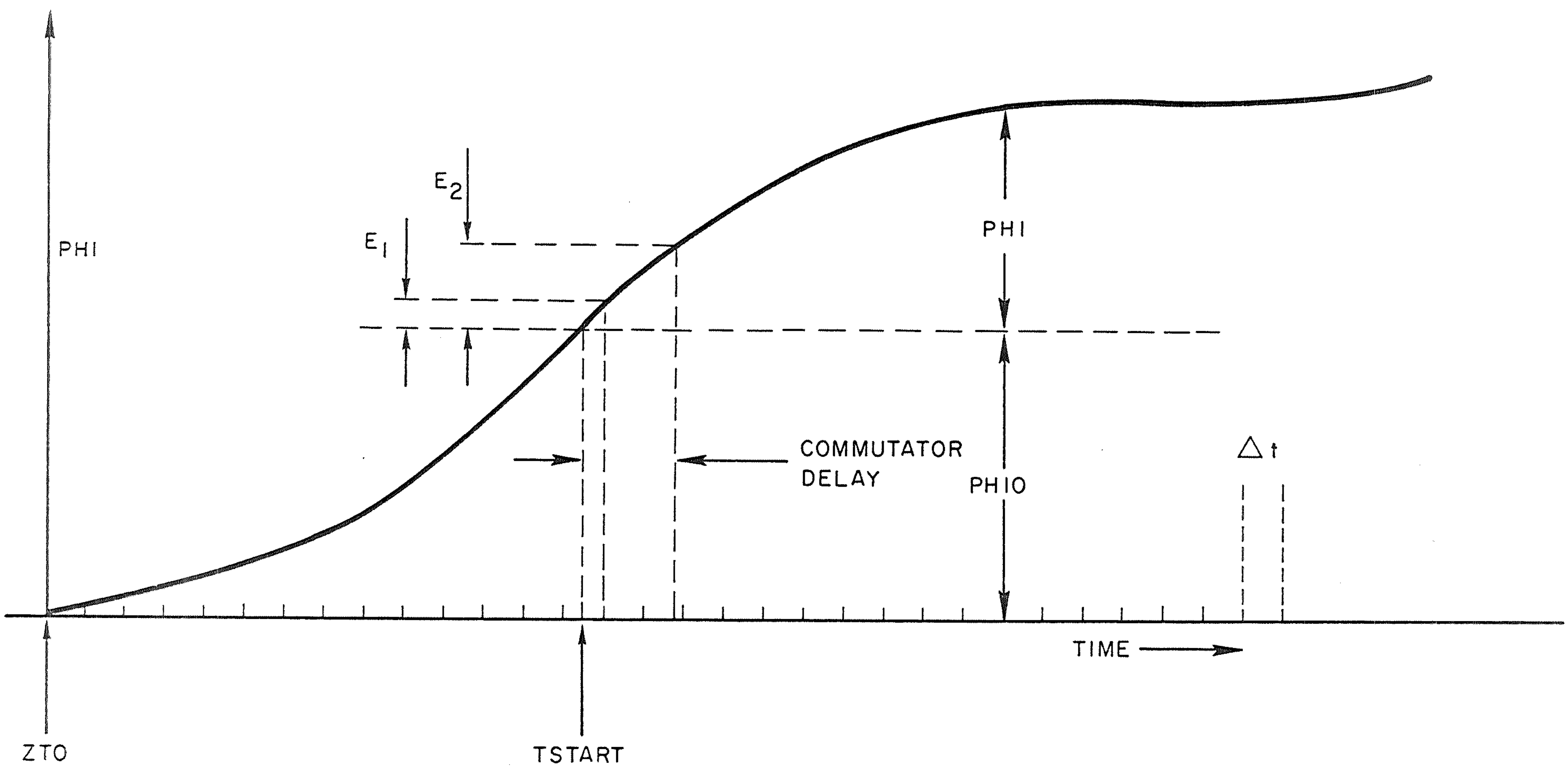


Figure 17. Definition of Precession Angle PHI
and Related Quantities.
See Module GADSSS.

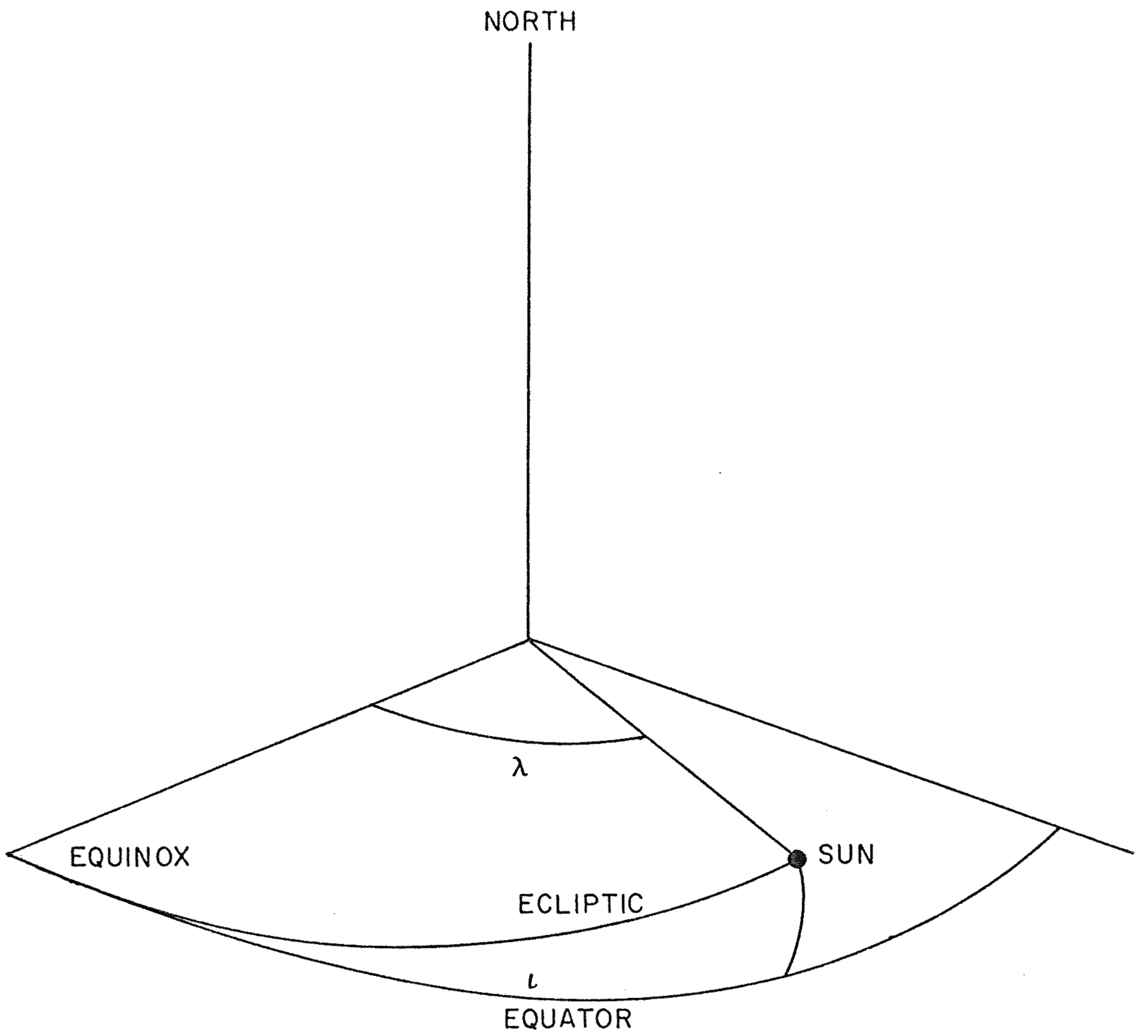
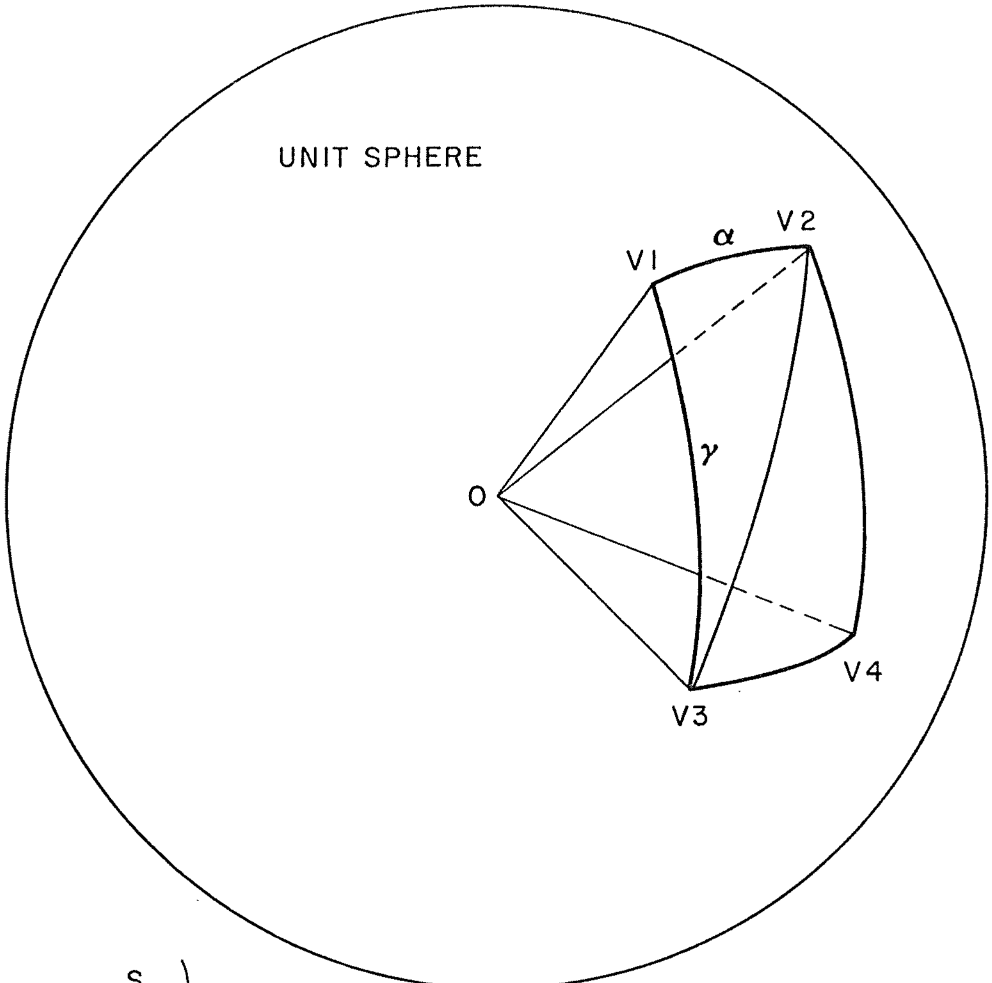
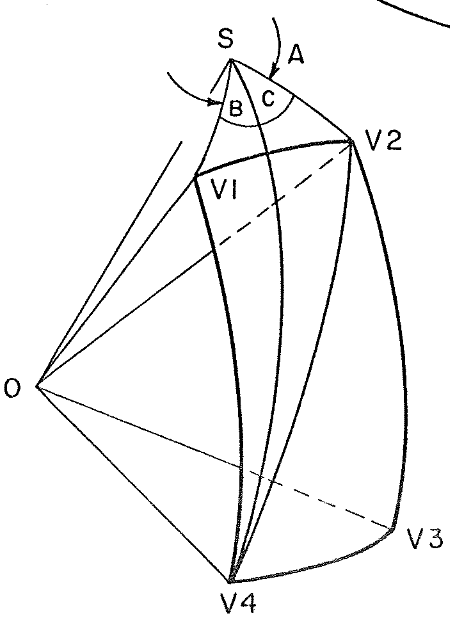
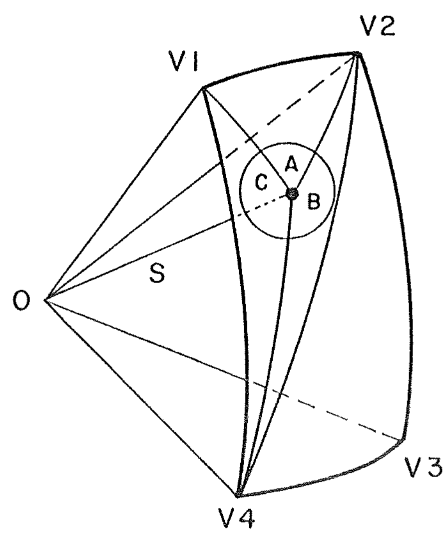


Figure 18. Longitude of Sun Referred to Mean Equinox. See Module GADSSE.



19 a





19b
NO SHADOW

Figure 19. Geometric Shadowing. See Module SHADOW.

19c
SHADOW

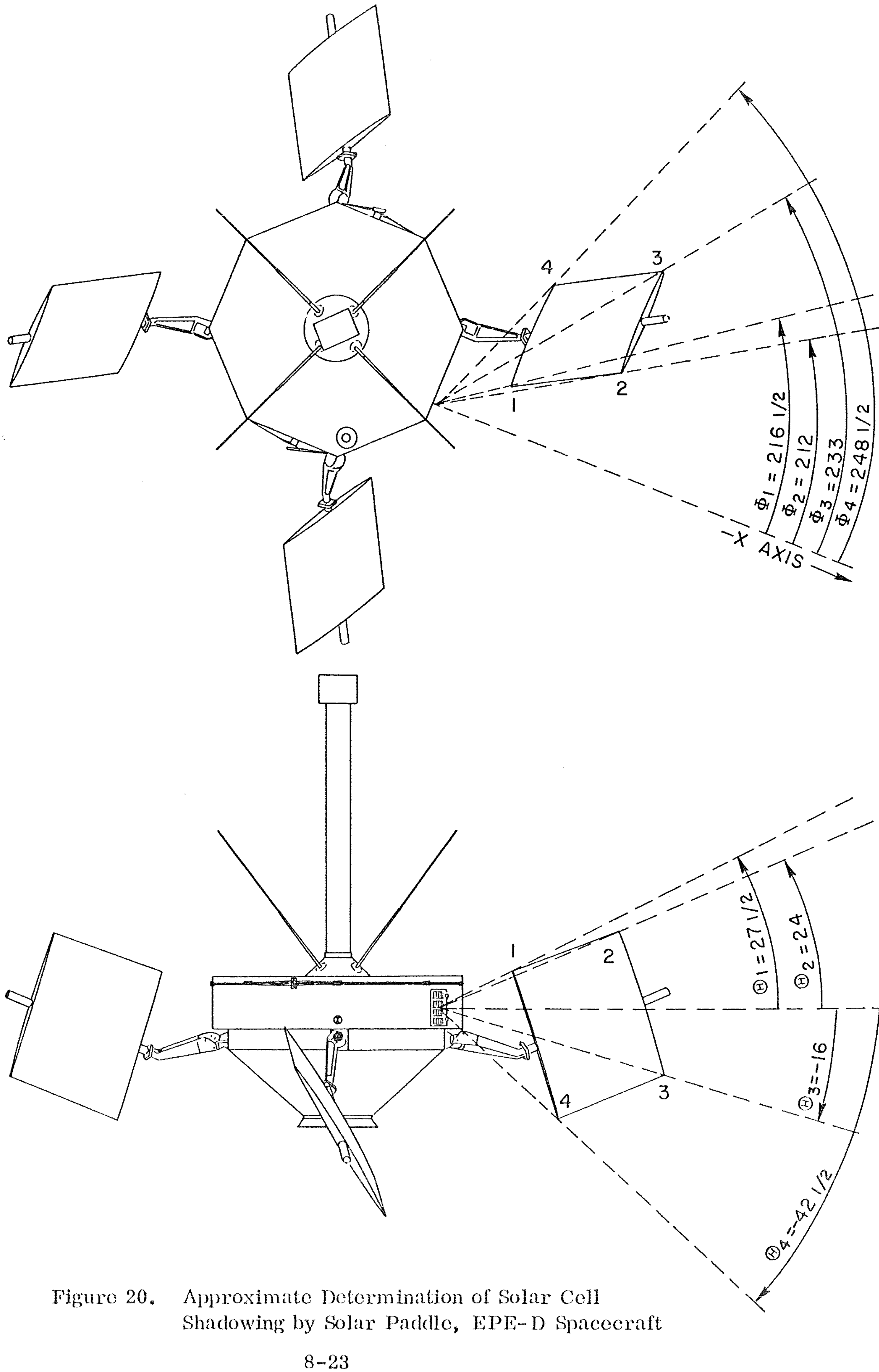


Figure 20. Approximate Determination of Solar Cell Shadowing by Solar Paddle, EPE-D Spacecraft

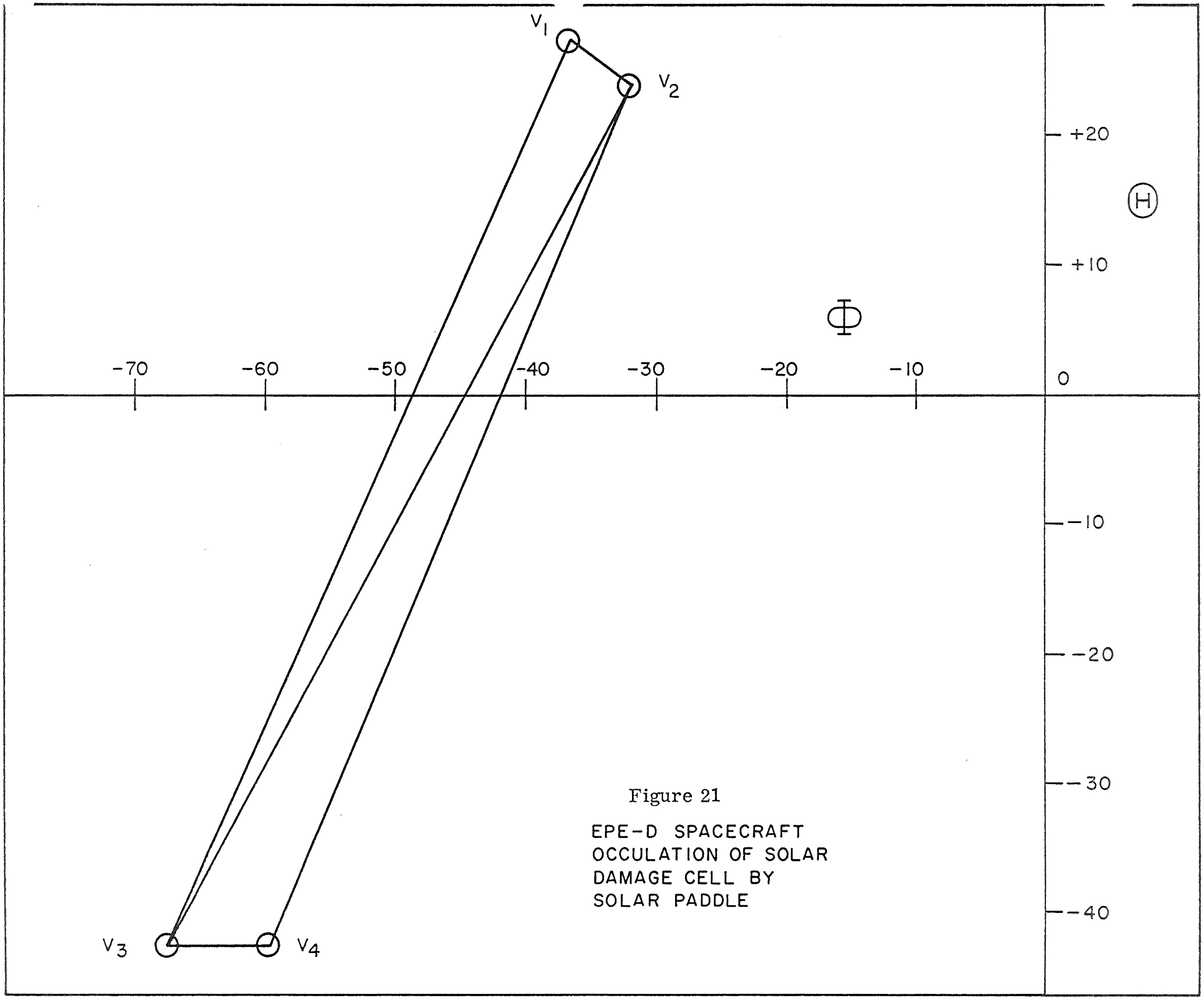


Figure 21

EPE-D SPACECRAFT
OCCULATION OF SOLAR
DAMAGE CELL BY
SOLAR PADDLE

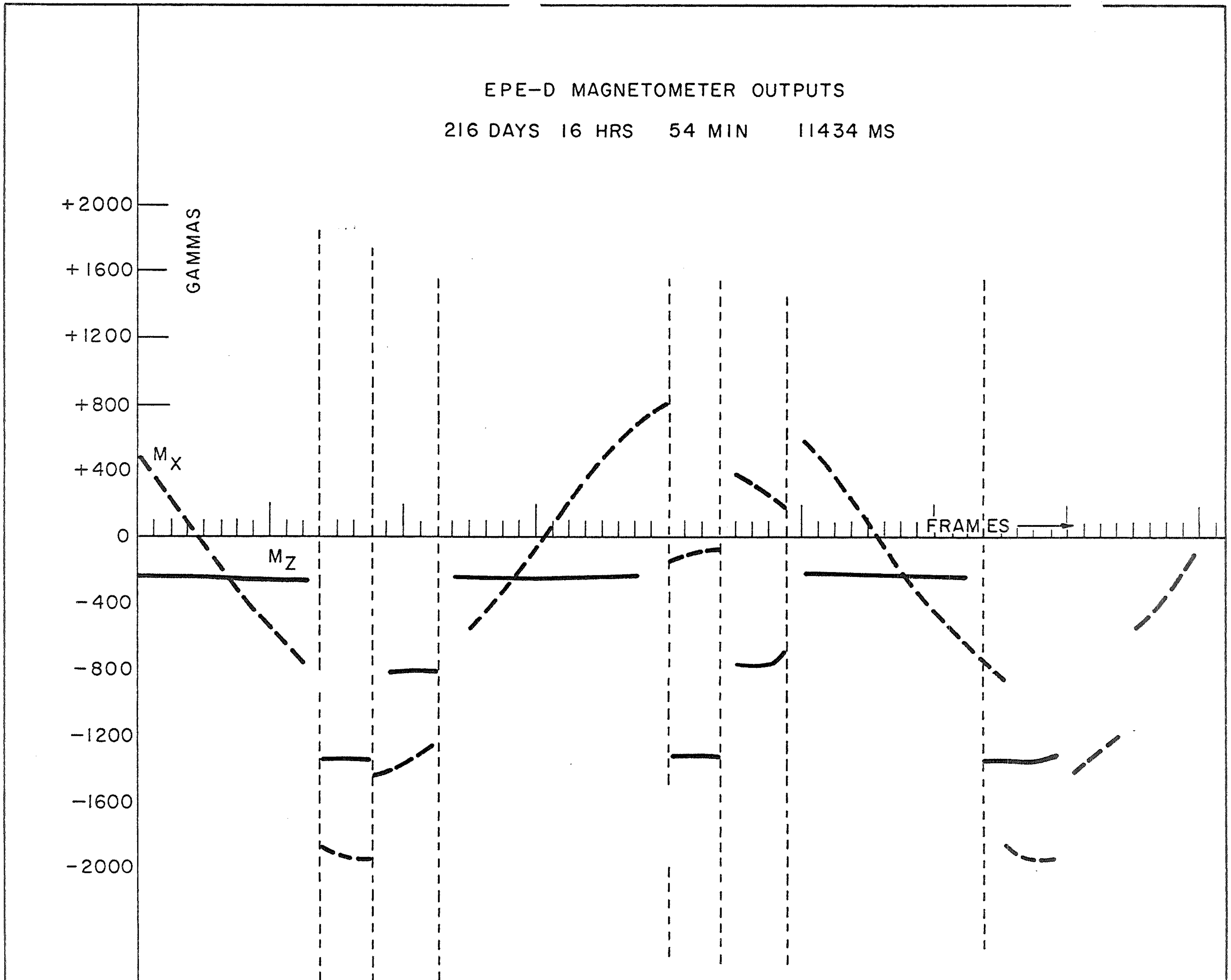

EPE-D MAGNETOMETER OUTPUTS
216 DAYS 16 HRS 54 MIN 11434 MS
GAMMAS
+2000
+1600
+1200
+800
+400
0
-400
-800
-1200
-1600
-2000
M_X
M_Z
FRAMES

Figure 22

	Type of I/O Action				
Module Name	Printer	Cards	Tape	Drum	Comment
GADA01	A			A	Drum used for auxiliary storage.
GADS/EPED	N	N	N	N	Executive for EPE-D.
GADS/ISIS	N				Executive for ISIS-A
GADSAO	A		N	N	General purpose advance orbit file.
GADSAT	A				Attitude Generator
GADSDP	N				Diagnostic Print
GADSFC	A				Function Chain
GADSIN	N	N			Initialization
GADSLS	N				Least Squares
GADSSS	N			N	Simulation of spacecraft and orbit.
GADSST	N				Statistics
IOHAND	A	N	N	N	General purpose I/O.

N = Normal
A = Abnormal

Table 1. Table of I/O Action

KGTYPE	JDERIV	Derivative Computed	Comment
1	1	$\partial \mathcal{E} / \partial \alpha$	Derivatives of ideal attitude matrix with respect to right ascension, declination, and Euler angles.
	2	$\partial \mathcal{E} / \partial \delta$	
	3	$\partial \mathcal{E} / \partial \varphi$	
	4	$\partial \mathcal{E} / \partial \theta$	
	5	$\partial \mathcal{E} / \partial \psi$	
2	1	$\partial S / \partial \alpha$	Derivatives of sensor operand with respect to same.
	2	$\partial S / \partial \delta$	
	3	$\partial S / \partial \varphi$	
	4	$\partial S / \partial \theta$	
	5	$\partial S / \partial \psi$	
3	1	$\partial \mathcal{K} / \partial \Phi$	Derivatives of geometric operator with respect to raw mounting constants.
	2	$\partial \mathcal{K} / \partial \Theta$	
	3	$\partial \mathcal{K} / \partial \cdot$	
	.	.	
	.	.	
4	1	$\partial \mathcal{G} / \partial c_1$	Derivatives of calibration operator with respect to raw calibration constants.
	2	$\partial \mathcal{G} / \partial c_2$	
	3	$\partial \mathcal{G} / \partial c_3$	
	4	$\partial \mathcal{G} / \partial c_4$	
5	25		Special cases such as complicated analytical partials, variational partials, and quadratures.
	26		
	27		
	.		
	.		
	.		
	.		

Table 2. Table of Derivative Types

Program Variable	Incremented by Module	Comment or Reason
MCALLS (1)	GADSLS	Inhibits repetition of diagnostics.
MCALLS (2)	GADSTV	Zero causes initialization.
MCALLS (3)	GADSSE	Zero causes initialization.
MCALLS (4)		Free.
MCALLS (5)		Free.
MCALLS (6)		Free
MCALLS (7)	GADSAO	Zero causes initialization of FASTRAND file.
MCALLS (8)	GRID1V	Nonzero causes printing of a plot frame.

Table 3. Table of MCALLS Allocations

Symbol	Code	Symbol	Code
O	1	I	25
X	2	G	26
.	3	S	27
Y	4	W	28
+	5	Q	29
*	6	(	30
L	7	4	31
U	8	)	32
0	9	N	33
H	10	-	34
C	11	=	35
V	12	D	36
=	13	A	37
Z	14	B	38
,	15	E	39
(	16	F	40
.	17	M	41
*	18	6	42
.	19	R	43
(	20	1	44
-	21	2	45
0	22	3	46
1	23	7	47
*	24	8	48

Note: These symbols are used by the GADS plotting programs on the on-line printer. They conform as much as possible to those shown on page II-81, Reference 10.

Table 4. Table of Plot Symbols for On-line Printer

Module GADA01. (Auxiliary function 01)

1.0 Calling Sequence: CALL GADA01 (F, X).

2.0 Category: FORTRAN subroutine, GADS worker.

3.0 Purpose: To compute the Euler angles as a function of time for a rigid force-free spacecraft which may be balanced or nonbalanced and which may be spinning, precessing, or tumbling.

4.0 Variables:

4.1 Explicit Inputs:

X(32) Array of intended system parameters. They may be perturbed or unperturbed, true (back-up) or trial solutions, or Taylor or gradient solutions. Refer to Note 1, module GADSLS for a fuller explanation of these expressions. Note that this variable is an example of communication by interpretation. See variable PRODEF, module GADSIN. If a non-Eulerian method of definition is used, a module of this type, namely GADA0N, N = 2, 4, 5, 6, 7, or 8 may be used. Consult variable AUXNAM, GADSIN.

4.2 Explicit Outputs:

F(8) Array of output auxiliary functions. This module computes the five Euler angles α, δ, φ, θ, and ψ, in that order. Observe that it is the responsibility of each module of the type GADA01 to store its output variables beginning with the LAUX+1 cell of F. The variable LAUX is discussed in GADSAC, GADSDC, and GADSFC. It may become necessary, after this writing, to increase the dimension 8 to some larger value.

N. B. This array can be thought of as the state variable in the GADS system.

4.3 Intermediate Variables and Implicit I/O Variables:

A Input. First principal moment of inertia.

AAA Same as A.

ALPHA Input. Right ascension (radians) of the auxiliary reference. As explained in Reference 1, Section II. K, the angular momentum is generally used to define the auxiliary reference frame.

AOUTI Hollerith commentary for printing.

AOUTS Hollerith commentary for printing.

B Input. Second principal moment of inertia.

BBB Same as B.

C Input. Third principal moment of inertia.

CCC Same as C.

CONS Constant used with change of elliptic modulus. See Note 4.

DELT Input. Step size for Simpson integration. This variable should be chosen with care. Refer to Note 4.

DELTA Input. Declination of the auxiliary reference. See ALPHA.

DPSDOT $\partial \dot{\psi} / \partial \theta$. See equation V. 25a, Reference 1.

ERRMES Hollerith commentary. This message may occasionally be printed when a transformation of elliptic modulus is necessary. See Note 4.

JMAX FORTRAN parameter. First dimension of arrays PHIS, WORK, WORKT.

KMAX FORTRAN parameter. Second dimension of arrays PHIS, WORK, WORKT, XSAVE. This parameter should be set to NCUT+1 where NCUT is the largest number of false position calculations expected.

LAUX Input. See modules GADSFC, GADSAF, GADSDC. This counter indicates the number of occupied cells in the output array F. See Output Variables.

MCUT Input. See GADSFC. MCUT is the counter for perturbed or false position calculations in the method of false position.

MOTION Input. Type of motion:

MOTION = 1: simple spin motion

MOTION = 2: Eulerian precession

MOTION = 3: general non-balanced force-free motion. See Section VI. F, Reference 1.

NAUX Input. Total number of auxiliary functions computed in all modules of the type GADA01.

NAUXDI Input. This array shows the number of output functions generated by each module of this type. Thus,

NAUXDI(1) = n^{0} functions generated by GADA01

NAUXDI(2) = n^{0} functions generated by GADA02

NAUXDI(3) = n^{0} functions generated by GADA03 etc.

See module GADSDA and Run Deck, cards G11 and G14b.

NCUT Input. Total number of false position calculations. Thus, $0 \leq \text{MCUT} \leq \text{NCUT}$.

NONCE1	Input. See Figure 11.
NONCE2	Input. See Figure 11.
NONCE3	Input. See Figure 11.
NONCE4	Input. See Figure 11.
NGOTO	NGOTO = 1: normal NGOTO = 2: a transformation of elliptic modulus has occurred.
NWSAVE	FORTRAN parameter. The number of constants of the motion to be stored in array XSAVE.
OBLATE	(A-C)/C.
PHI	Output. First Euler angle, ψ.
PHIACC	The accumulated value of PHI using Simpson integration.
PHIDOT	Output. $\dot{\varphi}$.
PHIO	Input. φ at time = 0.
PHIS(I, CUT+1)	The array of PHI values. This table is generated by means of Simpson integration and referenced later via linear interpolation. The user may, if he wishes, increase the step size DELT and adopt nonlinear interpolation for savings in core space and time. See also JMAX and Note 3.
POINST	Inverse of TSNIOP.
PSI	Output. Third Euler angle φ.
PSIDOT	Output. $\dot{\psi}$.
PSIO	Input. ψ at time = 0.

RPD	Radians per degree.
T	Time, temporary parameter.
TEMP	Array of temporary storage for constants of the motion.
THETA	Input and output. Second Euler angle, θ.
TIME	Input. Time for which attitude required (milliseconds since time origin).
TLOWER	Refer to Note 3.
TORIGN	Input. Origin of time for current attitude calculations. See Input Variables GADSLS.
TSNIOP	A constant of the motion, MOTION = 2.
USPECL	Input and output. Temporary storage for useful quantities. The main purpose of this array is to communicate key variables to other programs. Note, for example, that it transmits TSNIOP, DPSDOT to modules GADG26 and GADG27. The user may similarly wish to transmit key variables to additional modules of his own when adding a module of the type GADD01 and GADA01. Note that only programs of this type may store data in USPECL.
WORK(I, MCUT+1)	Array of integrands for SIMP2.
WORKT(I, MCUT+1)	Array of sample times for same. This array will be deleted in the near future.
XSAVE(I, MCUT+1)	Storage space for constants of the motion which should not be recomputed except once per differential correction iteration. See Note 2a.

ZALPHA	See equation 61.8, Reference 4.
ZAP	See equation 62.1, Reference 4.
ZARG	Argument of the elliptic functions.
ZBETA	See equation 61.4, Reference 4.
ZBQ	See equation 62.1, Reference 4.
ZCR	See equation 62.1, Reference 4.
ZG	Square root of ZGSQ.
ZGAMMA	See equation 61.8, Reference 4.
ZGSQ	See equation 60.5, Reference 4.
ZINTGR	Temporary location for the quantity $\dot{\varphi}/G$, the integrand of SIMP2.
ZK	Modulus of the elliptic functions. See equation 61.6, Reference 4.
ZP	The x-component of the angular velocity.
ZPO	The initial value of ZP.
ZQ	The y-component of the angular velocity.
ZQO	The initial value of ZG.
ZR	The z-component of the angular velocity.
ZRO	The initial value of ZR.
ZSIGMA	See equation 61.6, Reference 4.
ZT	Kinetic energy of rotation. See equation 60.2, Reference 4.
ZTO	t_0. Time (milliseconds) when the angular velocity lies in the x, z-plane. Refer to Section E.1b, Reference 1.

5.0 Notes:

5.1 Equations of Motion:

As stated in Section VI. F, Reference 1, there are three distinct types of motion in the force-free case: simple spin, Eulerian precession, and general or non-balanced motion. (The type of motion is determined by the user. See Run Deck, card G14a.) The equations needed for calculating the Euler angles in these three cases are given by Sections IV. 4 and IV. 6, Reference 1.

5.2 Special Programming Problems:

(a) Constants of the Motion. Constants of the motion need only be computed once for each iteration of the differential correction loop. The variable NONCE2 is referenced for this purpose. See Figure 11. In computing derivatives by the method of "false position", however, it may be necessary to evaluate these constants several times. Consequently, the results in the array XSAVE(I, MCUT+1) are saved for quick reference. The variable MCUT is explained in module GADSFC.

(b) Efficiency. Because the observations from the various sensors usually cover similar intervals of time (sampled by a commutator), it is desirable, at the expense of considerable core storage, to save the most time-consuming calculations, namely the φ calculations. (They are obtained by means of a Simpson integration.) A table of φ values are stored in array PHIS(I, MCUT+1) where MCUT has the same significance as in module GADSFC. Likewise, the integrand values for this integration are stored in array WORK(I, MCUT+1). Should these tables be exhausted, their contents are recorded in auxiliary storage areas for possible recall. To avoid I/O action, it is necessary that the size of these arrays is determined to be adequate. Refer to FORTRAN parameter JMAX. Adequacy of size is guaranteed if the quantity JMAX*DELT is larger than the largest expected TIME in milliseconds. See intermediate variable JMAX. The variable DELT is discussed later.

5.3 PHI Calculations:

In the non-balanced case, the calculation of PHI involves equation IV.4f, Reference 1. This is done with the help of module SIMP2, an integrator based on Simpson's rule.

In order to apply this method, a table of integrands is prepared in array WORK. This table has a constant step size, namely DELT, and is generated only as far as necessary. That is, if TIME is the time for which a PHI value is required and J denotes the number of elements already stored in WORK, then TIME > J*DELT causes the table to be extended. Otherwise, the value of PHI is simply obtained by interpolation.

If DELT is too small or too large, a loss of accuracy results in module SIMP2. In the former case, moreover, the tables are quickly exhausted. Hence, DELT must be chosen with care.

The proper choice can be made with knowledge of the approximate size of the period of the integrand, $\dot{\varphi}$. In this program, this period, ZPHIP, is computed from the formula 4.27, Reference 5. The formula provides an approximate value for the precessional rate $\omega_p = \dot{\varphi}$. Hence, the period is:

$$\text{ZPHIP} = 2\pi/\dot{\varphi}. \tag{1}$$

Computing a time step size corresponding to 1/ZSTEPS of a period:

$$\text{DELT} = \text{ZPHIP/ZSTEPS} \tag{2}$$

For example, if ZSTEPS = 120., then DELT corresponds to 3^o of precessional rotation.

Note that the approximate formula, taken from Reference 5, breaks down as ZK approaches 1. This corresponds to the situation where the angular velocity vector strays far from the largest or smallest principal body axes. That is, this corresponds to dynamical instability and calls for special handling. For example, the equations of motion could be integrated using module GADD01.

TLOWER, normally zero, is the lower limit of integration. When the WORK area has been exhausted, the lower limit of integration is replaced by the last upper limit.

5.4 Change of Elliptic Modulus:

Because the modules CN, SN, and DN, which compute the elliptic functions, assume that the modulus k of the said functions is in the range [0, 1.0], it may be necessary to use the following formulas:

$$SN\ (s,k) = k\ SN\ (sk,\ 1/k), \qquad (1)$$

$$DN\ (s,k) = CN\ (sk,\ 1/k), \qquad (2)$$

$$CN\ (s,k) = DN\ (sk,\ 1/k), \qquad (3)$$

Refer to paragraph 16.11, Reference 3. Note that, in this reference, m and u are the same as k^2 and s, respectively.

Whether or not the modulus k (see intermediate variable ZK) is in the range [0, 1.0] depends on: 1) whether or not the square of the magnitude of the angular momentum, ZG, is less than or greater than the quantity 2.0 * ZT * B which is twice the kinetic energy of rotation multiplied by the intermediate moment of inertia, and 2) whether $A < B < C$ or $A > B > C$.

Module GADC01. (Calibration, type 1)

1.0 Calling Sequence: CALL GADC01.

2.0 Category: FORTRAN subroutine, GADS worker.

3.0 Purpose: To compute the predicted telemetry value corresponding to a given ideal sensor output. That is, to transform from engineering units to telemetry units. Derivatives are also computed.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

ATEMP	Current observed data value.
CTEMP	Predicted sensor output function transformed (calibrated) into telemetry counts.
DEL	Residual. DEL = CTEMP - ATEMP.
FTEMP	Predicted sensor output function in engineering units; i.e., not calibrated.
GTEMP	Predicted sensor output function derivative.
GTEMP1	See module GADSGC.
JDERIV	Selector index for derivative.
KGTYPE	Type of parameter being processed.
NCOEFF	Number of coefficients in $\mathcal{G}$ operator.
NOBDAT	Refer to this variable in module GADSSP.
OCOEFF	Observable (sensor) calibration coefficients used in the function $\mathcal{G}$. These constants are loaded from OBCOEF in module GADSSP.

POLYN — Temporary locations for polynomial terms. See Notes.

5.0 Notes:

5.1 Calibration: The type of calibration performed in this module is:

$$\tau = \sum_{i=1}^{m} c_i \; f^{i-1} \qquad (1)$$

See Reference 1, equation VI. 15. Thus the engineering value f is transformed into a telemetry count value τ. The program symbols for τ, c, m, and f are CTEMP, OCOEFF, NCOEFF, and FTEMP, respectively.

5.2 Derivatives:

When derivatives are required, KGTYPE is non-zero. This variable determines the type of derivative and is explained in module GADSGC. The derivative of τ with respect to parameter u is given by:

$$\frac{\partial \tau}{\partial u} = \frac{df}{du} \cdot \sum_{i=2} c_i \; (i-1) \; f^{i-2} \qquad (2)$$

The program symbol for the left-hand side is GTEMP. The value of df/du is retained in GTEMP1. See program GADSGC.

Finally, the derivative of τ with respect to a calibration constant c_i is given by: $\frac{\partial \tau}{\partial c_i} = f^{i-1}$.

Module GADC04. (Calibration, type 4)

1.0 Calling Sequence: Call GADC04.

2.0 Category: FORTRAN subroutine, GADS worker.

3.0 Purpose: To compute the predicted engineering value from a given raw telemetry count value. Derivatives are also computed.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables: See module GADC01.

5.0 Notes:

5.1 Distinction Between This Module and GADC01:

This module calibrates the raw telemetry value, ATEMP, instead of the engineering value, FTEMP. Hence, its function is the mirror image of the previous module. This module is provided if it is necessary to adjust the coefficients in the inverse calibration curve as discussed below.

5.2 Inverse Calibration:

This calibration is given by:

$$f = \sum_{i=1}^{m} c_i \ \tau^{\,i-1}. \tag{1}$$

Thus the raw data value τ is transformed into engineering units f. The meaning of the symbols and their program names are the same as in module GADC01.

5.3 Derivatives:

The derivatives of f are given by:

$$\frac{\partial f}{\partial u} = \text{see equation (3), module GADF01;}$$

$$\frac{\partial f}{\partial c_i} = \tau^{i-1}. \qquad (2)$$

5.4 Programming Consideration:

Notice that using inverse calibration implies that the residual is:

$$DEL = FTEMP - CTEMP, \qquad (3)$$

instead of:

$$DEL = CTEMP - ATEMP. \qquad (4)$$

Module GADD01. (Differential equations, type 1)

1.0 Calling Sequence: CALL GADD01 (F, X, NDUMMY).

2.0 Category: FORTRAN subroutine, GADS worker.

3.0 Purpose: To compute the time derivatives of the fundamental state vector; i.e., to compute $\{\dot{\Omega}_x, \dot{\Omega}_y, \dot{\Omega}_z, \dot{\varphi}, \dot{\theta}, \dot{\psi}\}$.

4.0 Variables:

. Explicit Inputs:

NDUMMY First dimension of F.

X(K) Intended system parameters which must include the boundary or initial conditions. See Notes.

4.2 Explicit Outputs:

F(I, J) This array corresponds to the array F discussed under 5.2 for module ADAMS. Module GADSCS allots core space in such a way that:

F(I, 1) = F module ADAMS, location LDEQF
F(I, 2) = D module ADAMS, location LDEQG
F(I, 3) = P module ADAMS, location LDEQP
F(I, 4) = E module ADAMS, location LDEQE
F(I, 5) = Z module ADAMS, location LDEQZ

where:

I = 1 corresponds to φ
I = 2 corresponds to θ
I = 3 corresponds to ψ
I = 4 corresponds to Ω_x
I = 5 corresponds to Ω_y
I = 6 corresponds to Ω_z

The user should note, however, that the address F(1,1) need not necessarily coincide with F(1), module ADAMS, when several perturbed sets of differential equations are being integrated. In such a case, F(1,1) may be any of the following: F(1), F(7), F(13),... Therefore module GADD01 must be called once for each complete set of six coupled differential equations. (This is done in module GADFC1). Module ADAMS is unaware of these proceedings. Note that modules of the type GADA01 are, likewise, called once for each set of outputs. The user need not be concerned with these details unless he contemplates studying dumps of F or making changes.

4.3 Intermediate and Implicit I/O Variables:

AAA	Input. First principal moment of inertia.
ADME	Input. Accuracy, module ADAMS.
BBB	Input. Second principal moment of inertia.
CCC	Input. Third principal moment of inertia.
CONS	Constants of the motion.
LAUX	Input. Number of cells in state vector already occupied by other modules of the type GADD01. See module GADSDC.
LD	PARAMETER to identify ADAMS D. See 4.2.
LE	PARAMETER to identify ADAMS E. See 4.2.
LF	PARAMETER to identify ADAMS F. See 4.2.

MCUT	Input. Perturbation counter. See module GADFC1.
NONCE2	See Figure 11. This variable is an initialization flag and is turned off by GADSAL.
PO	Input. See notes.
QO	Input. See notes.
RO	Input. See notes.
TO	Input. See notes.
PHI	φ
PHIO	φ at time TO.
THETA	θ
THETAO	θ at time TO.
PSI	ψ
PSIO	ψ at time TO.
WX	See notes.
WY	See notes.
WZ	See notes.

5.0 Notes:

5.1 Differential Equations of Motion:

The differential equations describing the motion of a rigid body about its center of mass, in terms of Euler angles, are:

$$\dot{\varphi} = \left(\Omega_x \sin\psi + \Omega_y \cos\psi\right)/\sin\theta \qquad 1$$

$$\dot{\theta} = \Omega_x \cos\psi - \Omega_y \sin\psi \qquad 2$$

$$\dot{\psi} = \Omega_x \sin\psi \cot\theta - \Omega_y \cos\psi \cot\theta + \Omega_z. \qquad 3$$

$$a\dot{\Omega}_x = \Omega_y \ \Omega_z \ (b-c) + M_x \qquad 4$$

$$b\dot{\Omega}_y = \Omega_z \ \Omega_x \ (c-a) + M_y \qquad 5$$

$$c\dot{\Omega}_z = \Omega_x \ \Omega_y \ (a-b) + M_z \qquad 6$$

See page 36, Reference 1.

The present module provides the user with a convenient method of applying the driving terms M shown at the right of equations 4, 5, and 6. Hence, the task of studying a spacecraft subject to torques is, in principle, reduced to constructing an adequate force model, namely M.

5.2 Initial Conditions:

The initial conditions are references whenever NONCE2 = 1, which occurs 1+ NCUT times at the beginning of each differential correction loop. The initial values for the state vector are PO, QO, RO, PHIO, THETAO, PSIO. Note that good estimates of PO, QO, RO may require some preliminary calculations before calling the GADS system. See initialization Section of GADSSS for an example. In addition, note that initial conditions are usually being adjusted by differential correction.

Module GADF01. (Ideal sensor output function, type 1)

1.0 Calling Sequence: CALL GADF01.

2.0 Category: FORTRAN subroutine, GADS worker.

3.0 Purpose: To compute the ideal output function and derivatives of a "cosine" sensor.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

DEDU	Input. The derivatives of the transformation $\mathcal{E}$ with respect to α, δ, φ, θ ψ. See module GADSAL.
E	Input. The transformation to body coordinates $\mathcal{E}$.
FTEMP	Output. Sensor output function.
GTEMP	Output. Sensor output function derivative.
JDERIV	Input. Selector index for derivative.
JSCODE	Input. Selector index for sensor operand S.
KGTYPE	Input. Type of parameter being processed.
OGAMMA	Input. Direction cosines of sensor axis with respect to body coordinates.
OGAMMD	Input. Direction cosines differentiated with respect to "mounting constants". See module GADSR1.
S	Input. Sensor operand vector. See module GADSSO.
SPRIME	Input and output. Sensor operand vector S expressed in body coordinates.

5.0 Notes:

5.1 "Cosine" Sensor:

A sensor having an output f described by the vector dot product:

$$f = K \cdot S' = S \cos (K, S') \tag{1}$$

is called a cosine sensor. In this equation, K is a unit vector collinear with the sensor's sensitive axis in body coordinates. (The program symbol is OGAMMA). S is the environmental vector phenomenon upon which the sensor operates while S' is the same quantity expressed in body coordinates. Thus,

$$S' = \mathscr{E} S. \tag{2}$$

The program symbols for S, S', and $\mathscr{E}$ are S, SPRIME, and E, respectively.

5.2 Derivatives of f:

As discussed in Reference 1, Section V. F, the derivatives needed to perform differential correction can be separated into various parts. See equations V. 22, 23. In this module, interest is in all the derivatives except those concerned with the operator $\mathscr{G}$ Hence, the following is computed:

$$\frac{\partial f}{\partial u} = \left(\frac{\partial K}{\partial u}\right) \cdot S' + K \cdot \left(\frac{\partial S'}{\partial u}\right) \tag{3}$$

$$\frac{\partial S'}{\partial u} = \left(\frac{\partial \mathscr{E}}{\partial u}\right) \cdot S' + \mathscr{E} \left(\frac{\partial S}{\partial u}\right). \tag{4}$$

The derivatives of K are designated by the program symbol OGAMMD, those of $\mathscr{E}$ by DEDU. The symbol JDERIV selects the desired derivative. On the other hand, the various terms appearing on the right-hand side of equations (3) and (4) are processed according to the type of parameter. This is determined by the variable KGTYPE. For further clarification of these variables, see Table 2.

6.0 Programming Considerations:

6.1 Specialization:

Notice that this module is one of the highest working levels special to attitude determination. If it were not for the calibration (or transfer function) $\mathcal{G}$, this module would be the last stage in computing the predicted observed functions using: a) the information (i.e., the state vector) as provided by auxiliary functions (GADA01, GADD01), and b) the independent information provided by GADSSO. Hence, this module performs a similar role as those which compute range, range rate, azimuth, etc., in orbit determination.

6.2 Efficiency:

This type of module is invoked inside the main data loop and should be programmed efficiently.

Module GADFC1. (Function chain submodule 1)

1.0 Calling Sequence: CALL GADFC1 (PROGRM, AFAREA, NDUM).

2.0 Category: FORTRAN subroutine, GADS executive, level 3.5.

3.0 Purpose: To cause calculation of perturbed state vector (auxiliary functions) for use in numerical derivatives.

4.0 Variables:

4.1 Explicit Inputs:

PROGRM — Module to be used to call auxiliary functions:

PROGRM=GADSAC — closed form functions like the type computed in GADA01;

PROGRM=GADSDC — integrated functions like the type computed in GADD01.

NDUM — First dimension of AFAREA. See below.

4.2 Explicit Outputs:

AFAREA(NDUM, NCUT+1) — Computed perturbed auxiliary functions. For example, AFAREA(I, J+1) contains the Jth perturbation of the Ith auxiliary function. Note that AFAREA(I, 1) always contains the unperturbed functions. For greater detail concerning the method of perturbations, refer to 5.0 Notes.

4.3 Intermediate and Implicit I/O Variables:

CUTD — Perturbation determined by user. See Run Deck.

JUMPAL — "Jump GADSAL" flag. GADSAL computes the transformation $\mathscr{E} = \mathscr{A}\mathscr{S}$ which can sometimes be skipped during numerical differentiation.

KU — System parameter pointer index.

LINKEJ	Input. See module GADSIX.
LINKRF	Input. See module GADSIX.
MCUT	Perturbation counter.
N	Number of perturbations to compute on either side of "center". See Notes.
NDORDR	Input. Same as N. Also see module GADSIN and Run Deck, cards G9-1, 2, 3, 4 and G14c-1, 2.
NU	Input. Total number of active parameters.
SIGN	+1.0 and -1.0 flip-flop.
U(KU, 1)	Input. System "intended" parameters.
U(KU, 2)	Perturbed system parameters.

5.0 Notes: In order to compute perturbed sensor output functions, it is first necessary to compute perturbed auxiliary functions. This module is designed to cycle through all system parameters so as to determine when numerical differentiation is requested, to compute the auxiliary function perturbations on either side of "center", and to store the results in AFAREA.

Perturbations are stored sequentially using MCUT as a counter. Hence, there is no way for identification later (in modules GADFC3 and GADSGC) except by the same scheme used in their generation.

Module GADFC2. (Function chain submodule 2)

1.0 Calling Sequence: CALL GADFC2.

2.0 Category: FORTRAN subroutine, GADS executive, level 3.5

3.0 Purpose: To cause the calculation of: a) the transformation to body coordinates $\mathscr{E}$ (GADSAL), b) the ideal sensor output function f (GADF01, etc.), and c) the calibrated function τ (GADC01, etc.).

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

CTEMP	Output.	Calibrated sensor output.
FTEMP	Output.	Ideal sensor output, engineering units.
KSF	Input.	See module GADSSP.
KSO	Input.	See module GADSSP.
KSO1	Input.	See variable NOBDAT(3) in module GADSSP.

Module GADFC3. (Function chain submodule 3)

1.0 Calling Sequence: CALL GADFC3(AUXG, ADMP, LDUM, MDUM).

2.0 Category: FORTRAN subroutine, GADS executive, level 3.5.

3.0 Purpose: To cause calculation of perturbed predicted sensor output functions for use in numerical derivatives.

4.0 Variables:

4.1 Explicit Inputs:

AUXG(LDUM, NCUTS+1) Computed auxiliary functions. See variable AFAREA, module GADFC1.

ADMP(MDUM, NCUTS+1) Computed auxiliary functions obtained by means of module ADAMS.

LDUM First dimension of AUXG.

MDUM First dimension of ADMP.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

CTEMP Predicted sensor output function.

CTEMP1 Unperturbed predicted sensor output function. See module GADSFC.

CUTD See module GADFC1.

JUMPAL See module GADFC1.

KU See module GADFC1.

LINKEJ See module GADFC1.

LINKRF See module GADFC1.

MCUT See module GADFC1.

N See module GADFC1.

NDORDR	See module GADFC1.
NU	See module GADFC1.
PERTUR	Output. Perturbed predicted sensor output functions used in module GADSGC for calculating derivatives.
SIGN	See module GADFC1.
U	See module GADFC1.

5.0 Notes: This module is identical with GADFC1 except that it uses the results obtained to cause the calculation of the predicted sensor outputs. These are stored in PERTUR. The unperturbed sensor output resides in CTEMP1 as placed there by module GADSFC.

Modules GADG26, 27, 39, 40.

1.0 Calling Sequence: CALL GADG26, etc.

2.0 Category: FORTRAN subroutine, GADS worker.

3.0 Purpose: To compute the ideal derivatives of a "cosine sensor" output with respect to θ, $\dot{\varphi}$, and $\dot{\psi}$ for a dynamically balanced rigid spacecraft subject to no drag.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

DEDALF Input. $\partial \mathscr{E} / \partial \alpha$.

DEDDEL Input. See module GADSAL.

DEDPHI Input. See module GADSAL.

DEDPSI Input. See module GADSAL.

DEDTHE Input. See module GADSAL.

DEDU EQUIVALENT to DEDALF.

DPSDOT Input. $d \dot{\psi} / d \theta = - \dot{\varphi} \sin \theta \, (a-c)/c$.
See equation V.25a, Reference 1, and module GADA01.

GTEMP Output. Temporary location for ideal derivative.

III Term pointer and counter in the chain rule for differentiation. See Notes.

JDERIV Derivative selector.

JDERVS Library of derivative selectors.

OGAMMA Input. Fast access area for sensor direction cosines.

ORBVEC	Input. See module GADSSO.
POINST	Input. $c/(a-c) \cos \theta$. See module GADA01.
TSNIOP	Input. $d\,\dot{\psi}/d\,\dot{\varphi} = (a-c) \cos \theta / c$. See equation V.25b, Reference 1, and module GADA01.
USPECL	Input. Special system parameters. This array may contain special purpose and useful quantities. The basic purpose is to transmit variables from one working level to another and to reduce unnecessary recomputation of frequently used quantities. See module GADA01 for an example.

5.0 Notes: These modules exemplify the "non-standard" type of differentiation. The derivatives computed are non-standard because the chain rule is required. They are as follows:

Module	Derivative	Equation Numbers in Reference 1
GADG26	$\mathscr{E}\,\theta$	V.24d
GADG27	$\mathscr{E}\,\dot{\varphi}$	V.24f
GADG39	$\mathscr{E}\,\dot{\varphi}$	V.24c and V.26a
GADG40	$\mathscr{E}\,\psi$	V.24e and V.26b

Note that module GADSIN will automatically arrange for these modules without the user's intervention. He can, of course, override. See Run Deck and Usage.

Module GADGO1. ($\mathcal{G}$ operator number 1)

1.0 Calling Sequence: CALL GADGO1.

2.0 Category: FORTRAN subroutine, GADS executive, level 4.

3.0 Purpose: To load (prepare) the calibration operator (or transfer function) $\mathcal{G}$ when some of its parameters are "system parameters"; i.e., when some of its constants are being refined by differential correction.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

I	Pointer index for calibration constants (constants appearing in the "transfer function").
J	Pointer index for locating the said constants in the array of system parameters U.
LINKCP	Input. See module GADSSP.
KU1	Input. KU1 is usually set to 1. When KU1 = 2, "perturbed system parameters" are being used for a false position derivative.
OCOEFF	Output. Fast access area for calibration constants.
U(J, KU1)	Input. Array of system parameters. Here U corresponds to UTEMP in module GADSLS.

5.0 Notes: This module is used only when it is necessary to adjust (refine) the calibration constants of a given sensor. Under these circumstances it is clear that the quantites involved will be assigned places in the array of system parameters as described in Run Deck, cards G3, G5, and G14d. When these control cards are used, GADGO1 is automatically invoked to "update" the $\mathcal{G}$ operator as the differential correction progresses.

It is possible that the user may require a more complicated module which does more than merely move quantities. In such a situation, it may be necessary to compile a module starting with GADGO1 as a model. If a new module is produced (GADGO2), it may be invoked as required by placing the calling statement in module GADSSP. For example, if the new module is to be used only when processing sensor 6HSENSOR (6 hollerith characters used to identify the sensor), the following code could be placed in module GADSSP:

```
I=KXOR(SENSID, 6HSENSOR)
IF(I. NE. 0) CALL GADGO1
IF(I. EQ. 0) CALL GADGO2.
```

Module GADKO1. ($\mathscr{K}$ operator number 1)

1.0 Calling Sequence: CALL GADKO1.

2.0 Category: FORTRAN subroutine, GADS executive, level 4.

3.0 Purpose: To load (prepare) the geometric operator $\mathscr{K}$ when some of its parameters are system parameters; i.e., when some of its parameters are being refined by differential correction.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

I	Pointer index for geometric constants (constants appearing in the geometric operator).
J	Pointer index for locating the said constants in the array of system parameters U.
LINKOP	Input. See module GADSSP.
KU1	Input. KU1 is usually set to 1. When KU1=2, perturbed system parameters are being used for a false position derivative.
N6	Type of geometric mounting.
NOBDAT	Input. See variable NOBDAT under modules GADSSP and GADSGC.
OMOUNT	Output. Fast access area for geometric constants.
U(J,KU1)	Input. Array of system parameters. U corresponds to UTEMP in module GADSLS.

5.0 Notes: This module is used only when it is necessary to adjust (refine) the geometric or mounting constants of a given sensor.

Under these circumstances it is clear that the quantities involved will be assigned places in the array of system parameters as described in the Run Deck, cards G4 and G5. When these control cards are used, GADKO1 is automatically invoked to update the $\mathcal{H}$ operator as the differential correction progresses.

It is possible that the user may require a more complicated module which does more than merely move quantities. In such a situation, it may be necessary to compile a module starting from GADKO1 as a model. If a new module is produced (GADKO2), it may be invoked as required by placing the calling statement in module GADSSP. For example, if the new module is to be used only when processing sensor 6HSENSOR (6 hollerith characters used to identify it), the following code could be placed in the module GADSSP:

```
I=KXOR(SENSID, 6HSENSOR)
IF(I. NE. 0) CALL GADKO1
IF(I. EQ. 0) CALL GADKO2.
```

Module GADLS1. (Least squares submodule number 1)

1.0 Calling Sequence: CALL GADLS1 (A, NDUMMY).

2.0 Category: FORTRAN subroutine, GADS executive, level 2.

3.0 Purpose: To provide printout for module GADSLS only.

4.0 Variables:

4.1 Explicit Inputs:

A	Double precision square array to be printed.
NDUMMY	Dimension of A.

4.2 Explicit Outputs: See below.

5.0 Printout: The square array A(I,J) is printed with the format D15.10 such that the columns are printed horizontally.

Module GADLS2. (Least squares submodule number 2)

1.0 Calling Sequence: CALL GADLS2 (A, Y, DU, NDUMMY).

2.0 Category: FORTRAN subroutine, GADS executive, level 2.

3.0 Purpose: To provide printout for module GADSLS only.

4.0 Variables:

4.1 Explicit Inputs:

A	Double precision square array to be printed.
Y	Double precision vector to be printed.
DU	Double precision vector to be printed.

4.2 Explicit Outputs: See below.

5.0 Printout: The final printout is intended to display the following matrix equation:

$$Y = A*DU$$

This equation is in the form of the normal equations where Y is the gradient vector, A is the normal matrix of coefficients, and DU is the quasi-Taylor differential correction vector or solution.

Module GADS/EPED.

1.0 Calling Sequence: None.

2.0 Category: FORTRAN main program, GADS executive, level 0.

3.0 Purpose: EPE-D satellite attitude determination.

4.0 Variables:

4.1 Explicit Inputs: See 4.3.

4.2 Explicit Outputs:

4.3 Important Card Inputs:

Card 1. Format 16I5.

KBALF	See modules GADSSO and GADSSO/EPED.
KBDEL	See modules GADSSO and GADSSO/EPED.
KBTOT	See modules GADSSO and GADSSO/EPED.
KBTIME	Not used.
KOTAPE	See modules GADSSO and GADSSO/EPED.
NSETS	See modules GADSSO and GADSSO/EPED.
KWORDS	See modules GADSSO and GADSSO/EPED.
IBM	See modules GADSSO and GADSSO/EPED.
NITEMS	See modules GADSSO and GADSSO/EPED.

Card 2. Format 16I5.

LOCA	See module GADSCS.
LOCT	See module GADSCS.
LOCEND	See module GADSCS.
KSX	See modules of the type GADSSO.

KSY See modules of the type GADSSO.

KSZ See modules of the type GADSSO.

MSDORB See modules of the type GADSSO.

KDAYOR See modules of the type GADSSO.

KYEARO See modules of the type GADSSO.

KHOURO See modules of the type GADSSO.

Card 3. Format 16I5.

MDEBUG Card columns 1-5. Debugging aid:

0 = normal
1 or 3 = print words 45 through 300 of each input tape record
2 or 3 = print words 1 through 44 of same

NDRUMW Card columns 6-10. Drum words (FASTRAND) needed for storing orbit data. See Reference 19.

MPARAM(3) Not used.

MPARAM(4) Not used.

LMAX Card columns 21-25. See module GADSSS.

NCASE Card columns 26-30. See module GADSSS.

ICOND Card columns 31-35. See module GADSSS.

NDEBG1 Card columns 36-40. See module GADSSS.

KSIM Card columns 41-45. KSIM = 1 for simulation; KSIM ≠ 1 for processing EPE-D attitude. See Note 1, below.

KEOF1 Card columns 46-50. Not used.

KEOF2 Card columns 51-55. Not used.

NPROC	Card columns 56-60. Number of calls to GADSLS requested. See Note 1.
MF	Card columns 61-65. See module GADSSS.
NF	Card columns 66-70. See module GADSSS.
MORBIT	Card columns 71-75. See module GADSSS.
NORBIT	Card columns 76-80. See module GADSSS.

<u>Card 4. Format 16I5.</u>

NRECS	Card columns 1-5. Number of tape input records to process per call to module EPEDIN.
MPARAM(18)	Card columns 6-10. Not used.
NFRAMS	Card columns 11-15. Number of frames contained in one input record. See Reference 18.
NCHANS	Card columns 16-20. Number of channels per frame.
NPDPTS	Card columns 21-25. Minimum number of solar patch data points for attitude determination.
MWORDS	Card columns 26-30. Not used.
NWORDS	Card columns 31-35. Number of words per input record. See Reference 18.
MPARAM(24)	Card columns 36-40. Not used.
NBBCAL	Card columns 41-45. Number of consecutive frames criterion for determining the on or off state of the calibrating field. See Note 2.
NRSKIP	Card columns 46-50. Number of input records to skip after file is positioned.
NFSKIP	Card columns 51-55. Number of input files to skip.

NSAMPL Card columns 55-60. Sampling rate for magnetometer data, every NSAMPL will be processed.

IERRM Card columns 61-65. Maximum number of unsuccessful returns from GADSLS before run is terminated.

LEASTN Card columns 66-70. Least number of data points for attempt at attitude determination.

MIXERM Card columns 71-75. Not used.

NGREAT Card columns 76-80. Greatest number of data points that can be accepted for any sensor. This number is limited by the dimension of F1 and T1.

Card 5. Format 16I5.

MPARAM(33-40) Card columns 1-80. Not used.

Card 6. Format 8E10. 5.

XPARAM(1-8) Card colums 1-80. See module GADSSS.

Cards 7, 8, 9. See Card 6.

Card 10. Format 8E10. 5.

BBBCAL Card columns 1-10. Criterion for determining if the calibrating magnetic field has been turned on or off. Note that this number must be in counts or in gammas depending on whether the raw data is in counts or gammas, respectively. This, in turn, depends on whether module EPEDIN is calibrating the magnetometers.

BBBMAX Card columns 11-20. Criterion for determining when the geomagnetic field is too strong to attempt attitude determination (gammas).

DELTOR	Card columns 21-30. Delta orbit time. The largest data time span safe for attempting attitude determination, milliseconds.
PATCHS	Card columns 31-40. Solar patch slope. The slope m in the equations: $y = mx$ $x = y/m$ where y is in telemetry counts computed in module EPEDIN. Thus x is the sensor observation to be used as raw data in module GADSLS.
PATCHM	Card columns 41-50. Solar patch maximum acceptable telemetry value or count.
XPARAM(38)	Card columns 51-60. Initial value for Marquardt's mixture parameter. See variable XLAMIN, module GADSLS. This overrides module GADSDA.
XPARAM(39)	Card columns 61-70. Maximum for Marquardt's mixture parameter. See variable XLAMBM, module GADSLS. This overrides module GADSDA.
XPARAM(40)	Card columns 71-80. Not used.

<u>Card 11. Format 8E10.5.</u>

COMMUT	Card columns 1-80. See module GADSSS.

<u>Card 12. Format 8F10.4.</u>

TORIGN(1-7)	Card columns 1-80. See Usage and module GADSLS.

<u>Card 13. Format 4D10.5.</u>

DELT	Card columns 1-10. See module EPEDIN.
TFIXED	Card columns 11-20. See Usage and module GADSLS.
	Card columns 21-80. Not used.

Card 14. Format 16I5.

IREQST Card columns 1-80. See module EPEDIN.

Card 15. Format 16I5.

ICALB Card columns 1-15. See module EPEDIN.

NCOEF Card columns 16-30. See module EPEDIN.

Card 16. Format 16I5.

INTEP2 Card columns 1-40. See module EPEDIN.

ITEST Card columns 41-60. See module EPEDIN.

Card 17. Format 8E10.5.

EPCOEF Card columns 1-80. See module EPEDIN.

Card 18. Format 8E10.5.

EPCOEF Card columns 1-40. See module EPEDIN.

Card 19. Format 8F10.4.

AVERTX Card columns 1-80. See module SHADOW.

Card 20. Format 8F10.4.

SHADCR(1-4) Card columns 1-40. See module SHADOW.

Card 21. Format 8F10.4.

SIMSTA Card columns 1-80. See module GADSSS.

Card 22. Format 10A6.

EPREEL Card columns 1-6. EPE-D Attitude and Orbit tape described in module GADSSO/EPED and in Reference 18. These hollerith characters will be printed using the on-line typewriter to indicate the tape number to the operator. See Reference 19.

ORBFN	Card columns 7-12. FASTRAND file name for GADS use only. Any six characters legal to module IOHAND will do. See Reference 19.

4.4 <u>Intermediate and Implicit I/O Variables:</u>

AVERTX	Input. See module SHADOW.
BBBCAL	Input. See card 4.
BBBMAX	Input. See card 4.
COMMUT	Input and output. See card 11.
COMM	Input. See module GADSLS.
DELT	Input and output. See module EPEDIN.
DPR	Degrees per radian.
EPCOEF	Input. See card 18.
EPREEL	Input. See card 22.
F	Temporary area for transmission of raw data. While residing in F, the data is preprocessed.
F1	Terminal area for raw data.
FILLFL	Fill-data flag.
IBM	Input. See card 1.
IBUFEP	Buffer pointer. See Reference 19.
ICALB	Input. See card 15.
IEPSIZ	Buffer size for EPE-D input tapes.
IERRM	Input. See card 4.
IERROR	Counter for abnormal returns from GADSLS.
INSYNC	Not used.

INTEP1(8)	Eight cells used in EPEDIN. See variables K, KK, KKK, KKKK, etc.
INTEP2(8)	Input. See card 16.
INTEP3(8)	Eight cells used in EPEDIN. See variables I, II, III, IIII, etc.
IOBUF	Buffer pointer for orbit records on FASTRAND. See Reference 19.
IOBUF1	Buffer flip-flop pointer.
IOFLAG	See Reference 19.
IOFLEP	See Reference 19.
IOSIZE	Buffer size for orbit records. See Reference 19.
IREQST	Input. See card 14.
IRESET	See input variable NCALL in module GADSSS.
ISTART	Not used.
IT	Data pointer index.
ITEST	Input. See card 16.
KBALF	Input. See card 1.
KEOF	File marker counter, input tapes.
KEOF1	Not used.
KEOF2	Not used.
KEOT	End-of-tape counter, input tapes.
KPATCH	EQUIVALENT to INTEP2(7). Solar patch cell selector. See Note 4.
KSIM	Input. See card 3.
KWORDS	Input. See card 1.

LEASTN	Inpu . See card 4.
LMAX	Input. See card 3.
LPATCH	FORTRAN PARAMETER. Column indicator for storing solar patch data.
LOGIC	Type of I/O action. See Reference 19.
MCALLS	Call counters. MCALLS(7) is used by module GADSSO. See Table 6.
MDEBUG	Input. See card 3.
MF	Input. See card 3.
MIXERM	Input. See card 4.
MORBIT	Input. See card 3.
MPARAM	Input. See cards 3, 4, 5.
MWORDS	Input. See card 4.
NBBCAL	Input. See card 4.
NBRECO	Number of buffered records, orbit data on FASTRAND.
NCHANS	Input. See card 4.
NCOEF	Input. See card 15.
NCOMM	EQUIVALENT to COMM.
NDEBG1	See variable NDEBUG, module GADSSS.
NDRUMW	Number of drum words needed to store orbit data. See card 13.
NEWBDP	New B-field data point counter.
NF	Input. See card 3.
NFNR	Number of samples per input record per channel.

NFSKIP	Input. See card 4.
NGREAT	Input. See card 4.
NITEMS	Input. See card 1.
NMAG	Number of magnetometer samples per magnetometer.
NONCEX	EQUIVALENT TO INTEP2(8). See module EPEDIN.
NORBIT	Input. See card 3.
NPASS	Number of passes processed in EPEDIN.
NPASS1	NPASS mod 7.
NPATCH	Number of data points from solar patch.
NPDPTS	Input. See card 4.
NPROC	Input. See card 3.
NRECS	Input. See card 4.
NRESET	Number of words to clear when initializing arrays F1 and T1.
NRSKIP	Input. See card 3.
NSAMPL	Input. See card 4.
NSETS	Input. See card 1.
NSUN	Number of solar aspect sensor data points.
NWORDS	Input. See card 3.
ORBFN	Input. See card 22.
PATCHM	Input. See card 10.
PATCHS	Input. See card 10.
SHADCR	Input. See card 20.
SIMSTA	Input. See card 21.

STO	STOPT	Output. Stop time for a given attitude determination pass.
	T Samp,	Sample times corresponding to F.
	T1	Samples times corresponding to F1.
	TDATA	Input area for tape data.
	TFIXED	Input. See modules EPEDIN and GADSLS and Usage.
	TORIGN	Input. See card 12.
	VERTEX	See module SHADOW.
	XPARAM	See cards 6, 7, 8, 9, 10.

5.0 Notes:

5.1 Usage:

(i) This module reads input data from cards to control modules GADSSS, SHADOW, GADSLS and EPEDIN. Hence, the user is responsible for providing the necessary data and the input to module GADSIN.

(ii) This module reads EPE-D orbit-attitude tapes generated by the system reported in Reference 19. This tape is assigned by utilizing card 22.

(iii) When KSIM $\neq 1$, simulation of EPE-D is understood. To carry out simulation, however, the named common area ORBIT must be dimensioned as it is in GADSSS and GADSSO. The module GADS/SIM is provided for convenience.

(iv) This module will write an output attitude tape via module EPEOUT for which the user should provide an assignment card (Unit D).

(v) The user should consult Section 2 for additional usage requirements.

5.2 Magnetometer Data: Because magnetometers are susceptible to saturation and are subjected periodically to calibration, their outputs must be preprocessed. Saturated magnetometer data can be avoided by ignoring all observations taken when the spacecraft is in a field exceeding BBBMAX. On the other hand, the presence of a calibration field can be inferred by a jump in the recorded field strength exceeding BBBCAL gammas. Since the calibration is never "on" in excess of NBBCAL frame times, discard all observations bracketted by two sufficiently large breaks and few frames. In the program, refer to "DO 119 IT = 1, N10." (See Figure 22.)

5.3 Solar Aspect: As in module GADS/ISISA, solar aspect is obtained in degrees. Hence, it must be converted to cartesian components. See "DO 112 IT = 1, NFNR."

5.4 Solar Patch (Damage Cell): The solar damage cell experiments are in channel 15, frames 3, 4, 5, 6. That is, in PP3, 4, 5, 6. The user determines which cell is to be processed by means of input variable KPATCH and should be set to 3, 4, 5, or 6, respectively. KPATCH is read from card 16. Normally the input data will be transformed (calibrated) from telemetry counts (in the range 000-999) to a cosine. This can be performed in the preprocessor EPEDIN. In this instance, the user should punch +1.0 for variable PATCHS. See card 10. Alternatively, should the user want to obtain the raw telemetry counts from EPEDIN for reduction to a rough cosine in this module, he may do so by punching the appropriate normalizing factor in PATCHS and relieving module EPEDIN from this responsibility. Calibration in EPEDIN is controlled by variable ICALB, card 15.

A third alternative is to allow the GADS system to perform calibration. See Usage.

5.5 <u>Shadowing of Solar Patch</u>: Because the solar patch is subject to geometric shadowing from the solar paddle, the user should use the GADS modules GADF02 and SHADOW. The inputs to the latter module are read from cards 19 and 20. Existing constants were obtained by approximate methods. See Figures 19, 20, and 21.

5.6 <u>Sampling Magnetometer Outputs</u>: Magnetometer output is obtained every frame, and is sampled by means of the variable NSAMPL, card 4. Every NSAMPL frame will be sampled. See "DO 129 IT = 1, NEWBDP, NSAMPL".

5.7 <u>EPE-D Run Listings</u>: A listing of the card input for a typical EPE-D run is shown on the following two pages.

```
1XQT GADMAP
   15    16    14    18    35    63 1128   000    24
    1 102512000    21    22    23    18    19    20
    156000     3     4     6     2     3     1     1     0     1     1   256     6    24   047
    4     2    32    16    61    23   32643023     6   000     5    16     2
    2
+.00000+00+.60000+02+.57296+02+.000+00  +.000+00  +.37400+01+.40000+01+.54300+01
+.10000-00+.000+00  +.00000+00+.000+00  +.000+00  +.000+00  +.000+00  +.10000-01
+.10000+01+.00000+00+.40000+01+.000+00  +.000+00  +.000+00  +.000+00  +.000+00
+.20000+05+.000+00  +.30000+02+.19650+04+.20000+00+.10000+01+.000+00  +.72971-08
+.25000+03+.15000+04+.72000+07                    +.10000-03+.10000+00+.10000-03
+.000+00  +.000+00  +.000+00  +.000+00  +.000+00  +.000+00
   0.0        1.0        0.0          0.0        0.0         1965.0      0.0           0.0
+.10000+02+.00000+00
    3     6     5     1     1     0     0     0     0     0     0     0     0     0     0     0
    1     1     0     2     2     2
    2    16    61     2   104     3     5     0     4     2     6     0
-.50700+03+.55000+01+.00000+00+.00000+00-.50500+03+.58000+01+.00000+00+.00000+00
-.15000+03+.10000-02+.00000+00+.00000+00-.50500+03+.58000+01+.00000+00+.00000+00
  216.5       27.5       212.0       24.0          240.0       -40.0       248.5       -42.5
   355.0      69.0         0.0        0.0
  0.0         0.02       0.0           2.0        0.0          2.0
YYYYYYORBITE
START GADS CARDS
GADS RUN DECK
PARAMETER CALIBRATIONS
COEFF1 EPEDMX 1      COEFF2 EPEDMZ 1      COEFF3 EPEDMZ 2

PARAMETER NAMES
ALPHA     DELTA     PHIO      THETAO    PSIO      PHIDOT    PSIDOT    AAA
BBB       CCC       PO        QO        RO        TO        COEFF1    COEFF2
COEFF3

PARAMETER ESTIMATES
+.10000+00+.10000+01+.20000+00+.10000+01+.00000+00+.11000-00+.10400-00+.37400+01
+.40000+01+.54300+01+.54300+01+.37400+01+.45840+02.000000+00+.10000+00+.00000+00
-.10000+01-.10000+01-.10000+01
-.10000+01-.10000+01-.10000+01
OBSERVABLE(SENSOR)
SOLERR SOLAR  COSINE           1   256     0     0     0+.00000+00+.00000+00+.00000+00
```

```
027025037 +.12800+03+.30000+01-.10000+01+.10000+01
FIXEDV     +.18000+03+.00000+00
POLYN.     +.10000+03+.90000+03
+.10000+02
OBSERVABLE(SENSOR)
EPEDMY MAGNET COSINE         385  256    0     0     0+.00000+00+.00000+00+.00000+00
  2 45 38 +.12800+03+.30000+01-.40000+03+.40000+03-.10000+04+.10000+04
FIXEDV     +.45500+02+.00000+00
POLYN.     +.50700+03+.55000+01
+.10000+01
OBSERVABLE(SENSOR)
EPEDMZ MAGNET COSINE         641  256    0     0     0+.00000+00+.90000+02+.00000+00
 14 46 11 +.12800+03+.30000+01-.40000+03+.40000+03.110000+04+.10000+04
FIXEDV     +.00000+00+.90000+02
POLYN.     +.50500+03+.58000+01
+.10000+01
COMPLEX OF SENSORS
MAG2           RETURN                    018    0     0     0     2          +.30000+01
EPEDMZ    0EPEDMY    0SOLEPE    0

ALPHA      DELTA      PHIO       THETAO     PSIO       PHIDOT


CARRY AUXILIARY FUNCTIONS
"EULANG


DISPLAYS
COMPLEX    MAG2       FINALS
EPEDMZ     EPEDMY     SOLEPE

APLOT      EPLOT
+.12000+03
DISPLAYS
DETAIL
          2         0         0         2         1        40         0         0
          2         0         0         2         1        40         0         0
+.12000+02+.40000+02  .00+00     .00+00    .00+00    .00+00    .00+00   .00+00
+.12000+02+.40000+02  .00+00     .00+00    .00+00    .00+00    .00+00   .00+00
STOP
```

Module GADS/ISISA. (ISIS-A attitude determination)

1.0 Calling Sequence: CALL GADS ($1, $2, MAXT, ATTIT).

2.0 Category: FORTRAN subroutine, GADS executive, level 0.

3.0 Purpose: To provide the calling sequences for the determination of the attitude of the ISIS-A spacecraft and the generation of output attitude arrays.

4.0 Variables:

4.1 Explicit Inputs:

$1	Error return in case attitude cannot be computed.
$2	Alternate return to obtain a different sampling rate.
MAXT	Total number of attitude items to be computed.
ATTIT	Array of attitude items. This array may be partially loaded or computed upon entry to GADS.

4.2 Explicit Outputs:

ATTIT	Array of attitude items to be computed.

4.3 Intermediate and Implicit I/O Variables:

COMM	Input. Work area containing observed data and sample times.
DARKNS	Hollerith identification of first "sensor complex" to be processed when the spacecraft is in the dark.
ECLIPS	Eclipse status indicator obtained from orbit data tape.
LOCA	Input. Relative address of first cell in COMM containing observational data.

LOCATE	Input. Locations of observed data arrays. Thus, LOCATE(1) = relative starting location of observed data for first sensor; LOCATE(2) = relative starting location of observed data for second sensor; etc.

The preceding relative starting addresses are all referred to the first cell of COMM. (See Figure 5.)

JTIME	Output. Address modifier used to calculate the address of an observation sample time. Thus, if COMM(L) is the observed data, then COMM (L + JTIME) is the corresponding sample time.
LENGTH	Input. Lengths of observed data arrays. Thus, LENGTH(1) = length of array of observed data for first sensor; LENGTH(2) = length of array of observed data for second sensor; etc.
LOCT	Input. Relative address of first sample time in array COMM.
NSETS	Output. Number of sets of orbital data. Module GADSSO/ISISA is the customer.
ORBIT	Input. Orbital data array also referenced by the aforementioned module.
PROGID	Program hollerith identification.
RPD	Radians per degree.
SUNLIT	Hollerith identification of first sensor complex to be processed when the spacecraft is in the sunlight.

5.0 Notes:

5.1 Attitude Determination: Attitude determination is the determination of the parameters of the motion. The least squares estimate of the parameters of the motion is obtained in module GADSLS. Upon successful (normal) return from GADSLS, the desired output attitude is obtained by module GADSAT which makes use of the results obtained in GADSLS.

5.2 Types of Processing: The orientation of the angular momentum vector is best determined when the spacecraft is in the sunlight. Processing occurs in two distinct modes: sunlight and darkness. Cases 1 and 2 call for different sensor complexes; SUNLIT and DARKNS, respectively. Additional explanations can be found under module GADSLS.

5.3 Conversion of Sun Aspect Data: Because sun aspect data is provided in degrees, they are converted to cartesian components. This is done by taking sines and cosines. For example, the z-component is the sine function.

5.4 Determination of the Momentum Vector: A special entry, GADS1, has been provided to determine the orientation of the angular momentum vector. This entry will allow the processing of optical aspect and Z-magnetometer data for the single purpose stated. No attitude output will be generated. The call should be:

CALL GADS1($, $, MAXT, ATTIT).

For best results, the raw data for sensors ASPECT and Z 600 should span as wide an interval of time as is possible within a single pass. Upon return, the right ascension and declination of the angular momentum vector will, if possible, be updated and the results printed. These results will then be stored for subsequent use until they are refined by the next CALL GADS1.

5.5 Usage: The following steps illustrate the usage:

1. Prepare data from sensors ASPECT and Z 600.
2. CALL GADS1($, $, MAXT, ATTIT).
3. Prepare data from all sensors.
4. CALL GADS($, $, MAXT, ATTIT).

Module GADSAC. (Auxiliary function chain)

1.0 Calling Sequence: CALL GADSAC(F,X).

2.0 Category: FORTRAN subroutine, GADS executive, level 4.

3.0 Purpose: To call the necessary auxiliary function subroutines, if any.

4.0 Variables:

4.1 Explicit Inputs:

X	Array of intended system parameters.

4.2 Explicit Outputs:

F	Auxiliary functions. Example, F could be the state vector.

4.3 Intermediate and Implicit I/O Variables:

LAUX	Auxiliary function counter.
LINKAU	Input. Chain of auxiliary function selectors. This array is loaded from NCAUX in module GADSLS. NCAUX, in turn, is loaded by module GADSIN according to the rules of Run Deck, cards G14B.
NAUXDI(K)	Input. Number of functions generated by auxiliary function subroutine number K.

5.0 Notes: When fewer than 8 modules are called, the end of a chain of auxiliary computations is signalled by the first zero element in LINKAU. Note the similarity with module GADSDC.

Module GADSAI. (Attitude item)

1.0 Calling Sequence: CALL GADSAI(ATTIT).

2.0 Category: FORTRAN subroutine, GADS worker.

3.0 Purpose: To calculate or transmit special output calculations to the output array.

4.0 Variables:

4.1 Explicit Inputs:

ATTIT	Array of attitude output. This array variable represents one attitude item, in contrast to the array of the same name in module GADSAT. This array is considered as an input and as an output because it may be partially computed in other modules.

4.2 Explicit Outputs:

ATTIT	Output area.

4.3 Intermediate and Implicit Input Variables:

B	Third Euler rotation matrix.
C	Second Euler rotation matrix.
D	First Euler rotation matrix.
E	Euler transformation; E = BCDR.
ORBVEC	Orbit-related environmental variables. See module GADSSO.
R	Rotation matrix from G. E. I. to auxiliary system. R corresponds to the $\mathcal{T}$ transformation in Reference 1.
U	System parameters. See module GADSLS, Note 1.

5.0 <u>Notes</u>: This module allows the user to reference key attitude variables which may be included in each output attitude item. For example, the body axes are given, in the G.E.I. system, by the rows of E. Likewise, the z-axis of the auxiliary reference (the angular momentum) is given by the last row of R. Statistical results can also be obtained through the array COMM. Refer to the calling sequence of GADSST from module GADSLS. When referring to ORBVEC, the user should remember that this variable will not have been loaded since the last call to GADSSO. To obtain updated results, the following instructions should be included in module GADSAT just above CALL GADSAI:

```
JSCODE=J
CALL GADSSO,
```

where J=1, 2, ... depending on whether the solar vector, magnetic vector, ... is desired.

Module GADSAL. ($\mathcal{A}\mathcal{S}$ transformation)

1.0 Calling Sequence: CALL GADSAL(X).

2.0 Category: FORTRAN subroutine, GADS worker.

3.0 Purpose: To compute the ideal coordinate transformation to body coordinates.

4.0 Variables:

4.1 Explicit Inputs:

X — State vector containing the five angles $\alpha, \delta, \varphi, \theta, \psi$ which must be stored in the first five locations of X in the order shown. These angles define the instantaneous orientation of the vehicle at TIME. This module is especially designed for use in the Euler method of parameterization. It is understood that:

X(1) = right ascension of auxiliary reference,

X(2) = declination of auxiliary reference,

X(3) = first Euler angle,

X(4) = second Euler angle,

X(5) = third Euler angle.

This is a typical example of "communication by interpretation". Hence, GADSAL is coded in conformity with the output variable F, module GADA01. Also see Run Deck, card G2. If the user desires to apply GADS to a problem defined in terms of roll, pitch, and yaw, for example, this module should be replaced by a similar module conforming to the new definition of the angles.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

ALPHA	Output. Right ascension angular momentum or other auxiliary reference. See Section II, Reference 1.
B(3, 3)	Output. Third Euler rotation matrix. See equation II.11a and II.12, Reference 1.
BC	Output. The matrix product of B and C.
BCD	Output. The matrix product B, C, and D. See equation II.12, Reference 1.
C(3, 3)	Output. The second Euler rotation matrix. See equation II.11b, Reference 1.
CD	Output. The matrix product of C and D.
D(3, 3)	Output. The first Euler rotation matrix. See equation II.c, Reference 1.
DBDPSI	Output. $dB/d\psi$.
DCDTHE	Output. $dC/d\theta$.
DDDPHI	Output. $dD/d\varphi$.
DEDALF	Output. $dE/d\alpha$.
DEDDEL	Output. $dE/d\delta$.
DEDPHI	Output. $dE/d\varphi$.
DEDPSI	Output. $dE/d\psi$.
DEDTHE	Output. $dE/d\theta$.
DELTA	Output. Declination of angular momentum vector or other auxiliary reference. See ALPHA.
DRDALF	Output. $dR/d\alpha$.

DRDDEL	Output. $dR/d\,\delta$.
DUMMAT	Dummy temporary matrix.
E(3, 3)	Output. The ideal coordinate transformation to body coordinates. See equation II. 9, Reference 1.
KCCUTS	Input. False position differentiation flag bits. See Note 4, GADSLS.
KOCUTS	See KCCUTS, above.
KSCUTS	See KCCUTS, above.
MOTION	Input. Type of motion indicator. See same variable, GADA01.
NONCE1	Input. Repetition inhibitor. See Figure 11.
NONCE2	Input and output. Repetition inhibitor. See Figure 11 and module GADA01.
PHI	Output. First Euler angle at TIME.
PSI	Output. Third Euler angle at TIME.
R(3, 3)	Output. Transformation from inertial to auxiliary reference. R corresponds to the $\mathfrak{L}$ transformation in Reference 1.
THETA	Output. Second Euler angle at TIME.
TIME	Current time in milliseconds referenced to origin.

5.0 <u>Notes:</u>

5.1 <u>Formulas:</u> The equations programmed in GADSAL correspond to those of section II, Reference 1. All non-vanishing derivatives with respect to the five angles α, δ, φ, θ, and ψ are also computed.

5.2 Type of Motion: Much computation can be exercised, depending on the type of motion. Thus:

MOTION = 1: only ψ changes with time,

MOTION = 2: only ψ and φ change with time,

MOTION = 3: only ψ, φ, and θ change with time,

MOTION > 3: all angles change with time.

For these reasons, the variable MOTION is interrogated.

5.3 False Position Differentiation: When false position derivatives are to be computed, it is not safe to assume that ALPHA and DELTA are constant. Therefore, the repetition inhibitor NONCE2 is not invoked in this case.

Module GADSAO. (Advance orbit tape)

1.0 Calling Sequence: CALL GADSAO.

2.0 Category: FORTRAN subroutine, GADS executive, level 5.

3.0 Purpose: To advance I/O device containing orbital and environmental data.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate Implicit I/O Variables:

IBM	Input.	IBM = 1: input data in IBM format; IBM = 0: input data in UNIVAC format.
K	Input.	$K \geq 0$: read forward; $K < 0$: rewind before reading.
KOTAPE	Input.	FORTRAN unit number of input device.
KWORDS	Input.	Number of words in one input record.
NCALLS	Call counter.	
ORBIT	Output.	Orbit data array.
ORBIT1	Dummy address.	

Module GADSAT. (Compute spacecraft attitude)

1.0 Calling Sequence: CALL GADSAT(TARRAY, ATTIT, ITEMS, ITEMST, MAXT).

2.0 Category: FORTRAN subroutine, GADS executive, level 1.

3.0 Purpose: To compute final fitted results, namely the attitude output as a function of time.

4.0 Variables:

4.1 Explicit Inputs:

TARRAY	Array of abscissae for which final results are desired.
ITEMS	First dimension of array ATTIT.
ITEMST	First dimension of array TARRAY.
MAXT	Number of abscissae in TARRAY and number of attitude output sets desired.

4.2 Explicit Outputs:

ATTIT	Array of attitude output.

4.3 Intermediate and Implicit I/O Variables:

6HOUTPUT	Hollerith flag used to identify a sensor complex suitable for output calculations.
ADMH	See module GADSML.
ATEMP	See module GADSML.
CTITLS	See module GADSLS.
IGNORE	See module GADSML.
IT	Pointer index for i sample time.
ITAGIT	Flag to cause the tagging of outliers. See module GADSLS.

JETRAN	Flag to obtain $\mathcal{E} = \mathcal{A}\mathcal{S}$ transformation only. Refer to module GADSFC.
JS	Sensor pointer index.
JSCODE	S code or sensor operand code.
KSO	Flag to request sensor calibration operator.
LAUXG	Refer to module GADSML.
LC	Sensor complex pointer index.
LDEQE	See module GADSML.
LDEQF	See module GADSML.
LDEQG	See module GADSML.
LDEQP	See module GADSML.
LDEQZ	See module GADSML.
LDEQ1	See module GADSML.
NAUX	See module GADSML.
NC	Total number of sensor complexes.
NCUT	See module GADSML.
NDEGF	See module GADSML.
NDEQ	See module GADSML.
NDEQ1	See module GADSML.
NONCE2, 3, 4	See module GADSML.
NTOTF	See module GADSML.
NTOTR	See module GADSML.
PROGID	Program identification.
SUMSQ	See module GADSML.

TIME	Elapsed time in milliseconds since TFIXED.
TFIXED	Millisecond of year (or launch) used as a time base for calculations. Double precision.

5.0 Notes:

5.1 Calculation of Final Results:

This module performs a function similar to that of GADSML. That is, it provides the necessary executive functions to calculate the predicted functions (attitude) using the abscissae provided in TARRAY instead of those belonging to the observed raw data. It also inhibits all unnecessary functions such as those connected with differential correction. The variable JETRAN is the said inhibitor.

The user can provide whatever special calculations desired within module GADSAI.

5.2 How the Sensor and Sensor Complex are Chosen: Since the modules which perform the prediction calculations do so with reference to specific sensors and sensor complexes, JS and LC must be assigned legitimate values. The former can be arbitrarily set to 1; the latter can point to any sensor complex. For most efficient use of computer time a sensor complex with the least number of calculations should be chosen, i.e., with the fewest number of parameters. Such a complex will have the flag 6HOUTPUT in the corresponding array CTITLS. See Run Deck, card 14-1. (When no such complex is found, LC is arbitrarily set to 1.)

5.3 How the Attitude Sample Times are Determined:

MAXT attitude items will be computed for the MAXT sample times (referred to TFIXED as in all other calculations).

These samples times are assumed to be in

$$TARRAY(1, J);\ 1 \leq J \leq MAXT.$$

For an example, refer to Note 5.1, module GADS/ISISA.

Module GADSCS. (Core space)

1.0 Calling Sequence: CALL GADSCS($).

2.0 Category: FORTRAN subroutine, GADS executive, level 2.

3.0 Purpose: To allot core storage work space.

4.0 Variables:

4.1 Explicit Inputs:

$	Alternate return if core storage requirements exceed resources.

4.2 Explicit Outputs: No explicit output variables.

4.3 Intermediate and Implicit I/O Variables:

COMM	Common work area to be partitioned. COMM may or may not be "named" common, but must be available to the entire GADS system throughout each call.
IFREE	Output. The surplus of core cells.
IQUIT	Output. The maximum number of iterations allotted to sensor complex LC.
JTIME	Output. Address of a data sample time relative to the data sample. In GADS, JTIME is assumed to be the same for all data.
LAUXG	Output. Relative starting address for auxiliary functions.
LC	Input. Pointer index for sensor complex under execution.
LDEQE	Output. Relative starting address of E array used in module ADAMS.

LDEQF	Same as LDEQE for array F.
LDEQG	Same as LDEQE for array G.
LDEQP	Same as LDEQE for array P.
LDEQZ	Same as LDEQE for array Z.
LDEQ1	Output. Total space needed for each of the preceeding arrays E, F, G, P.
LINKF	Dummy work space.
LOCA	Input. Relative starting address of first observed data.
LOCAIN	Output. Relative address of matrix inverse in solution of normal equations.
LOCAMA	Output. Relative address of first quasi-Taylor matrix of coefficients. See Figure 10.
LOCANR	Output. Relative address of normalized matrix of coefficients.
LOCAXY	Output. Relative address of unnormalized matrix of coefficients.
LOCDU	Output. Relative address of first quasi-Taylor vector.
LOCD1	Output. Relative address of first residual.
LOCEND	Input. Highest available address.
LOCFLG	Output. Relative address of flag array used in detection of outliers. See Note 3, GADSLS.
LOCF1	Output. Relative address of first predicted sensor output function.
LOCG1	Same as LOCF1 for first gradient.
LOCHF1	Same as LOCF1 for constraint functions.

LOCHG1	Same as LOCF1 for constraint gradients.
LOCNOR	Output. Relative address for normalization factors. See variable Y, module GADSNE and CNORM, module GADSST.
LOCONE	Output. Relative address of certain results obtained from module MATINV.
LOCT	Same as LOCA for corresponding sample time.
LOCTEL	Same as LOCF1 for predicted telemetry value; i.e., for calibrated sensor output. See equation II.6, Reference 1.
LOCTOP	Output. Highest address needed to execute sensor complex LC.
LOCV1	Output. Address of first V vector. See Figure 10.
LOCWRK	Same as LOCV1 for WRK vector.
LOCYMA	Same as LOCV1 for Y matrix.
LOCYNR	Same as LOCV1 for normalized Y vector.
LOCY1	Same as LOCV1 for Y1 matrix.
MCALLS(I)	Subroutine call counters.
NAUX	Output. Total number of auxiliary functions.
NCALLS	Output. Total number of times module GADSCS is called.
NDEQ	Output. Total number of unique differential equations.
NDEQ1	Output. Effective total number of differential equations when perturbed systems are taken into account.
NH	Output. Total number of constraints; i.e., multipliers.

NOCORE	Output. Scarcity of core space indicator: NOCORE = 0, no scarcity NOCORE = 1, scarcity This flag is referenced by modules GADSML and GADSNE.
N7RIES	Output. Maximum number of outlier detection trials. See Note 3, GADSLS.
NS	Output. Maximum number of sensors in complex LC.
NT	Output. Maximum number of data points allowed for any sensor within complex LC.
NTNS	Output. NT*NS.
NTNSNU	Output. NT*NS*NU.
NT1	Output. Permanent storage for NT. See module GADSML.
NT2	Output. This variable is used as a divisor during calculation of core storage requirements: NT2 = 1, no scarcity of core NT2 = NT1 = NT, scarcity of core storage
NU	Output. Maximum number of active system parameters exclusive of multipliers.
NUMC1	Input. Packed parameters related to sensor complex definition.
NUNH	Output. NU*NH.
NV	Output. NU+NH; i.e., total number of parameters to be adjusted.
NV1	Output. Total number of cells required for specific arrays in modules GADSNE and GADSMN.

NV1SQ	See NV1.
NV2	See NV1.
NV3	See NV1.
NV4	See NV1.
NZ	Output. Total number of sensors being calibrated in this complex.

5.0 Notes: With some loss of speed, considerable work core space can be relinquished if it is allotted in compact form. This module computes exact core requirements for each sensor complex. At first, the open core method is attempted. Should this prove excessive, the addresses are recomputed in the compact method. If this method should also fail, the error return is invoked.

The open method results in NOCORE = 0; the closed method results in NOCORE = 1. The open method has no advantage except during checkout stages if it is desirable to obtain core dumps of various arrays.

The amount of available core space is determined by the user who must supply this number in variable LOCEND before the first call to GADSIN. Refer to user instructions.

Module GADSDA. (Data pool)

1.0 Calling Sequence: Not applicable.

2.0 Category: FORTRAN block data, GADS executive, level 1.

3.0 Purpose: To pool constants and other data.

4.0 Variables:

ASTRSK	Hollerith asterisk for flagging refined parameters in printout.
AUXNAM	Hollerith names of specialized functions, i.e., auxiliary functions. See Run Deck, card G11.
CALIBP	Hollerith names of calibration constants acting as system parameters. Blanks are used to neutralize this array.
CRGAMM	Critical value for the angle gamma. See Note 1, module GADSLS.
DIAGN	Hollerith library of diagnostics used by module GADSLS.
DMILLS	Milliseconds per day.
DPR	Degrees per radian.
EPS	Library of convergence criteria. See Run Deck, cards G8.
FILLFL	Flag for identifying data fills.
ISTAND	Maximum number of standard derivative functions. See module GADSGC and Run Deck, cards G13-2, 3, 4, 5.
JSTAND	Maximum number of standard derivatives recognized by module GADSIN. Refer to Run Deck, cards G13-2, 3, 4, 5.

LAMBDM	Used in GADSLS: 1 = compute differential corrections for only one value of Marquardt's x λ. 2 = compute differential corrections for two values of x λ. See Note 1, module GADSLS.
LAMBIN	Initial value for LAMBDM. The user may wish to use LAMBIN = 1 or 2, depending on how well he knows the correct values of the system parameters. If in doubt, use 2. See also variable XLAMIN below.
NAUXDI(K)	The number of auxiliary outputs produced by the K auxiliary function, such as GADA01.
NBIT24	A useful bit configuration. See GADSTV.
ORIENP	Hollerith names of mounting or geometric constants acting as system parameters. Blanks serve to neutralize this array.
PI	Ratio of circle's circumference to its diameter.
POWERS	Useful bit configurations, the first 16 powers of 2.
RPD	Radians per degree.
SEMIPI	PI/2.
TLABEL	Hollerith labels corresponding to the contents of the array TORIGN. See Section 2, Usage.
TWOPI	PI*2.0.
UACC	Accuracy parameters. See Run Deck, cards F7.
UCUTS	Library of perturbations for calculating false position derivatives. See Run Deck, cards G9.

XKDUCR	Critical value for Marquardt's k. When $k < \text{XKDUCR}$, the gradient method fails to detect a gradient and it may be concluded that the gradient vanishes. See Note 1 and variable XKDU in module GADSLS.
XLAMBC	Critical value of Marquardt's x_{λ}. See module GADSLS.
XLAMBM	Maximum value for Marquardt's x_{λ}. See module GADSLS.
XLAMIN	Initial value of Marquardt's x_{λ}. See module GADSLS.
YMILLS	Double precision milliseconds per year.

5.0 Notes: The intention is to collect in this module all constants and other quantities which the user may question or modify. In this way changes are simplified. Note that most quantities may be changed after loading; i.e., during execution if desired, so that recompilation is obviated. Arrays ORIENP and CALIBP should not be disturbed since they affect the decision processes.

Module GADSDC. (Differential equation chain)

1.0 Calling Sequence: CALL GADSDC(F,X).

2.0 Category: FORTRAN subroutine, GADS executive, level 4.

3.0 Purpose: To call the modules which compute the time derivatives of the desired auxiliary functions.

4.0 Variables:

4.1 Explicit Inputs:

X	Array of intended system parameters.

4.2 Explicit Outputs:

F	Derivatives of auxiliary functions or the state vector.

4.3 Intermediate and Implicit I/O Variables:

LAUX	Auxiliary function counter.
LINKAU	Input. Chain of auxiliary derivative selectors. This array is loaded from NCAUX which is, in turn, loaded from the Run Deck, card G14f, by modules GADSLS and GADSIN, respectively.
NAUXDI(K)	Input. Number of derivatives generated by Kth auxiliary derivative function subroutine.

5.0 Notes: When fewer than eight modules are called, the end of a chain of auxiliary derivatives is signalled by the first zero element after LINKAU(8).

Note the similarity with module GADSAC.

Module GADSDP. (Diagnostic printout)

1.0 Calling Sequence: CALL GADSDP.

2.0 Category: FORTRAN subroutine, GADS executive, level 2.

3.0 Purpose: To provide generalized diagnostic printout.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

DIAGNO	Permanent array of hollerith diagnostics.
JSOS	Input. Diagnostic sub-type indicator or code number.
KSOS	Input. Diagnostic type selector.
KARD	Input. Last card pointer.
RUNDEK	Permanent array for run deck card images in hollerith.
SOS	Input. Name of calling program, hollerith.

Module GADSDV. (Data for displays)

1.0 Calling Sequence: Not applicable.

2.0 Category: FORTRAN block data, GADS high-speed printer display support only.

3.0 Purpose: To provide a centralized grouping of display constants needed only when using the high-speed printer in place of the SC 4020.

4.0 Variables:

BLANKS	Hollerith blanks for clearing print area.
BLANK1	Hollerith characters for left margin.
DASHES	Hollerith dashes for horizontal grid lines.
DASH1	Hollerith characters for left edge of grid lines.
EJNOT	Hollerith space suppressing characters.
JOPERS	1108 "J" operators for partial word handling. See Reference 8 and Figures 13a and b.
QUESTN	Hollerith symbols used to signal that the number of data points residing in a specific grid cell exceeds one.
SYMBLS	Hollerith symbols available for plotting. These conform, as much as possible, to those of Reference 10, page II-81, also shown in Table 4.
VERTIC	Hollerith I's for vertical grid lines.

Module GADSFC. (Function chain)

1.0 Calling Sequence: CALL GADSFC(AARRAY, FLG, AUXG, ADMF, ADMP, LDUM, MDUM).

2.0 Category: FORTRAN subroutine, GADS executive, level 3.

3.0 Purpose: To produce the calculation of predicted functions. When numerical differentiation is requested, this module also causes computation of all perturbations.

4.0 Variables:

4.1 Explicit Inputs:

AARRAY Raw (observed) data.

FLG The outlier detection flag. During an outlier detection trial (NTRIAL $\geq$ 0) and if the data was previously found undesirable, the appropriate bit will be on. For example, assume that the observed data AARRAY is an outlier and that the source of the data was the (j+1) th participating sensor (see Glossary). In such a case, the 2^j bit in FLG is 1.

LDUM First dimension of the output array AUXG. Therefore, LDUM is equal to the number of auxiliary functions.

MDUM First dimension of the output array ADMF. Therefore, LDUM is equal to the number of differential equations.

4.2 Explicit Outputs:

AUXG(i, j) This two-dimensional array contains the current auxiliary functions. The first index i refers to the function. During numerical differentiation, the jth perturbation of the ith function is found in AUXG(i, j+1). Moreover, AUXG(i, 1) is the unperturbed set of functions.

ADMF(i, j)	This array is similar to AUXG except that it contains the auxiliary derivative functions being numerically integrated in module ADAMS. See Run Deck, card G14f.
ADMP(i, j)	This array contains the interpolated values of the primitive functions of ADMF. (Note: The integrator ADMINT may be somewhat ahead of TIME; the integrator's current time is ADMT).

4.3 <u>Intermediate Variables:</u>

ADMT	The current time (milliseconds) in the integrator ADMINT.
ATEMP	Alternate location for raw data.
AUXF	Named common area for auxiliary functions including the integrated functions. This alternate location simplifies communication.
CTEMP	Predicted calibrated sensor output function.
CTEMP1	Predicted calibrated sensor output function used in module GADFC3.
CUTD	Temporary cell for perturbation.
FTEMP	Temporary predicted sensor output function, engineering units.
FILLFL	Fill-data flag. See module GADSDA.
IGNORE	Flag used to stop processing of a specific observed data.
JETRAN JUMPAL	A flag to signal when the "$\mathcal{AJ}$" or ideal transformation may be skipped. See Reference 1, Section II.

LINKRF	Input. Linkage array signalling "regula falsi" derivatives. Assume that this method of differentiation is applied to the KUth system parameter and the parameter occupies the Kth active position in the normal equations. Then, LINKRF(K)=KU.
KSAUX KSF	Ideal prediction function selector.
KSO	Non-ideal operator (calibration) request flag.
KSO1	Type of calibration selector.
KSW	Weighting selector.
KU	System parameter pointer.
KU1	KU1=1 during normal calibrations, and KU=2 during calculation of perturbed functions.
MCUT	Perturbation counter.
NAUX	Number of auxiliary functions.
NCUT	Total number of perturbations.
NDEQ	Number of differential equations being integrated.
NFILLS	Fill-data counters, one for each sensor.
NOBDAT	See module GADSSP.
NTOTF	Total number of fill-data counter.
NTOTR	Total number of rejected raw data.
NU	Total number of active parameters.
PROGID	Hollerith program identification.
TIME	Time since time origin, milliseconds.
UCUTS	Resident array for perturbations.

UCUTS1 Temporary array for perturbations.

WEIGHT Temporary location for the weight function. See modules GADSSP and GADW01 and Run Deck, cards G13-9, G14a-2, 3.

5.0 Notes:

5.1 General Comments:

Module GADSFC is concerned with calculation of predicted sensor output functions. This module is called once for each data point; i.e., for each sample time. To accomplish the required calculations, the spacecraft's attitude is predicted and placed in the 3 x 3 matrix E by module GADSAL. The pure parameters of the motion and TIME are used as inputs. Once E is obtained, the specific characteristics of the sensor and sensor operand are used. These are handled by modules of the type GADF01 and GADSSO. Furthermore, when false position differentiation is called for, GADSFC also computes the perturbed predicted functions with the help of modules GADFC1 and GADFC3. As explained in Reference 1, the (predicted) quantities being fitted to the observed data, generally, are functions of functions. The latter may, in turn, be functions of yet other functions, and so on. In order to reduce redundant calculations, these preliminary functions are carried separately as auxiliary functions. In this way, reference may be made later without redundant calculations. Auxiliary functions are computed in modules of the type GADA01 and GADD01.

5.2 Auxiliary Functions:

a) Standard: The auxiliary functions are all those which are required to construct the three operators $\mathcal{G}$, $\mathcal{H}$, and $\mathcal{E}$. See Section II. Reference 1. The most basic functions are the Euler angles which appear in the ideal prediction operator. In the force-free case, these angles are obtained by means of equations IV.4 or IV.6. In the forced case, it may be necessary to integrate equations IV. 3. The force-free cases are handled in module GADA01, the forced case in GADD01.

b) Nonstandard: The user may have reason to carry additional auxiliary functions. In this case, he is required to code his program(s) as GADA02, 3, ...8 and/or GADD02, 3, 4, ...8. To insure that these functions are computed, the user should follow instructions as outlined in Run Deck, cards 14b and 14f.

Note 1: All functions coded in modules of the type GADD01 are assumed to form a system of coupled first order differential equations. That is, the resulting auxiliary functions are the integrals or primitives of those coded in the said modules. Integration is automatically performed in module ADAMS.

Note 2: Any single auxiliary module may produce more than one auxiliary function. For example, GADA01 generates α, δ, φ, θ, and ψ, or a total of five functions. The number of functions generated must be found in NAUXDI. See modules GADSAC, GADSDC, and GADSIN.

5.3 Numerical Differentiation:

When certain derivatives are to be computed by the method of false position, the perturbed and unperturbed auxiliary functions are automatically generated. The exception to this rule occurs when a certain parameter does not affect the auxiliary functions. A perturbation in a calibration constant, for example, need not result in calculation of perturbed Euler angles. In such cases, a flag bit is inserted in the variable LINKEJ. Refer to module GADSIX. This bit is tested in module GADFC1.

Module GADSGC. (Gradient function chain)

1.0 Calling Sequence: CALL GADSGC.

2.0 Category: FORTRAN subroutine, GADS executive, level 3.

3.0 Purpose: To produce the calculation of all derivatives needed in differential correction.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

CUTD	Perturbation for use in numerical derivative.
DEL	Input. Weighted residual computed in module GADSFC.
GTEMP	Temporary location for derivative.
GTEMP1	Temporary location for ideal derivative; i.e., for the derivative of the ideal (noncalibrated) sensor output function with respect to parameters not affecting the calibration. Hence, GTEMP1 is given by formula V.22b, Reference 1.
IGNORE	Input. Ignore the current observed data flag set in module GADSFC. When data is ignored, the residual and associated derivatives are set to zero.
JDERIV	Derivative selector. See modules GADSIN, GADF01, GADC01 and Table 2.
KGTYPE	Type of parameter being adjusted. See Table 2.
KSG	Derivative module number. Normally KSG = KSF (module GADSFC). For nonstandard differentiation,

however, KSG is determined by JDERIV. See modules GADG26, 27, 39, and 40 for an example.

KSO	Input. Calibration request flag. See module GADSSP.
KSO1	Input. EQUIVALENT to NOBDAT(3). See module GADSSP.
KU	Active system parameter selector.
K1	Special flag:

K1 = 0 - process ideal derivative as described under GTEMP1,

K1 = 1 - process derivative with respect to calibration constants and geometric constants.

LINKG — Input. Derivative selectors.

LNKKG1 — Input. Parameter type selectors.

NDORDR — Input. Order of numerical differentiation; i.e., the number of perturbations on either side of center:

NDORDR(1) - number of perturbations for first "active parameter",

NDORDR(2) - number of perturbations for second "active parameter", etc.

Note: Since perturbations are computed on either side of center, twice NDORDR(K) perturbed values of the predicted sensor output function will result for the Kth parameter.

NU — Input. Total number of active system parameters.

PERTUR — Input. Array of perturbed functions computed in GADFC3. To understand the order of appearance of these perturbations as stored by the said module, assume that numerical derivatives are being computed for parameters u and v; these parameters are declared in Run Deck cards G14-4, 5, in that order; they occupy active positions KU = I and KU = J (I less than J), respectively; and NDORDR(I) = 2 and NDORDR(J) = 1. Then the contents of PERTUR are as follows:

PERTUR(1) = unperturbed function f(u, v)
PERTUR(2) = f(u+e)
PERTUR(3) = f(u-e)
PERTUR(4) = f(u+2e)
PERTUR(5) = f(u-2e)
PERTUR(6) = f(v+d)
PERTUR(7) = f(v-d)

where,

e = UCUTS1(I)
d = UCUTS1(J)

SENSID — Input. Hollerith identification of current sensor. See module GADSSP.

UCUTS1 — Input. Perturbations needed for numerical derivatives. See module GADSIX.

USREL1 — Input. System-parameter-to-sensor relationship, if any. See modules GADSIN and GADSIX.

WEIGHT — Input. Current weighting for data.

5.0 Notes:

5.1 Numerical Derivativatives:

Upon entry, the array UNCUTS1 contains non-zero perturbations for those parameters for which numerical derivatives are to be computed. These calculations are performed beginning with statement number 1001.

N, half the order of the numerical derivative, is contained in array NDORDR. Intermediate results are held in XRF and YRF which are the perturbed abscissae and ordinates, respectively. The final generalized formula is given just below statement 1008. The manner in which perturbations are generated is also discussed under Section 4.3, above. See variable PERTUR and Run Deck, card G9-1.

5.2 Flow Chart:

Further understanding of this module is provided by Figures 3a and b and Reference 1, page 62ff.

Module GADSIN. (Initialization and interpretation of user instructions)

1.0 Calling Sequence: CALL GADSIN($).

2.0 Category: FORTRAN subroutine, GADS special purpose initialization.

3.0 Purpose: To read, interpret, and translate the user instruction deck only once and to initialize the GADS system for attitude determination.

4.0 Variables: Because of the size of this module, it is desirable to discuss the processing in a descriptive way. Hence, we consider the coding in a semisequential manner. In the discussion which follows, the series of FORTRAN statement numbers 1, 2, ...32 and 401, 402, ...432 correspond to the 32 possible types of input controls. (See variable ENGLIS for the list of 32 controls.) The latter series of statement numbers (400 series) occur only during initial processing of the physical control cards. The former series refer to regular (post-initialization) processing after all cards have been read.

Control 1 - loads system parameter names into PARAMS, checks for implied parametrization, counts total number of parameters NU. If any or all of the first JSTAND fields are blank, they are interpreted as implied parameters. Each of these blank fields is replaced by the corresponding contents of PCODES containing the standard parameter names.

Control 2 - loads estimates of the system parameters.

Control 3 - loads accuracy of the estimates. If the estimate is perfect, then UACC should be 1.0; if the estimate is a guess UACC should be 0.

Control 4 - loads differential correction convergence requirements or criteria. During differential correction, changes (or fractional changes) in system parameters are compared with EPS in order to determine when satisfactory convergence has been achieved.

Control 5 - processes observable (sensor) cards:

1. Reads sensor title OTITLE, sensor operand code, sensor output function type, function code (if any), location and length of raw data in COMM. Variables İİ, J̇J̇, KK are provided if the user requires special-purpose codes which could be used in module SENSOR.

2. Records maximum length of raw data array, namely MAXNT0. This variable will determine amount of space needed for outlier detection.

3. Initializes standard derivative codes and replaces the codes according to user's instructions, if any. This occurs at Control 9.

4. Processes derivative data, including finding the parameter address and its type. Refer to Table 2. Computes the arrays NUMF and NUMG containing the sensor's output function code and derivative codes, respectively. Thus:

NUMF(J) = output function number, sensor J;
NUMG(n, J) = derivatives numbers, sensor J.

These derivative codes are packed six to a word.

5. Reads sensor display data.

6. Reads the sensor mounting type and the raw mounting constants which determine the geometry of the mounting.

7. Reads the sensor calibration type and the calibration parameters which determine the transformation of engineering units to telemetry counts or vice-versa.

8. Generates NOBDAT. The contents of these cells are given under module GADSSP.

9. Reads weighting factor OBWGHT. Note that this factor may be superseded if the user stipulates a weighting function under COMPLEX OF SENSORS.

10. Calls module SENSOR which transforms raw mounting constants into direction cosines and performs all other calculations of a once only type. Note that variable N6 communicates the type of geometry. The user can add special sensor calculations of his own in that module, if necessary.

Control 6 - loads constraint name, code, and function code, if any. Next the constraint's derivative codes are loaded. Constraints are discussed in Reference 1.

Control 7 - processes sensor complex data:

1. Loads sensor complex identification or name and special controls, number of iterations, special purpose codes NUMC, number of outlier isolation trials, ignore complex flag, type of motion, and tolerance level for outlier detection. The array NOBDAT is used for convenience as a dummy.

2. Loads names of participating sensors and the desired weighting function, if any.

3. Loads names of participating system parameters.

4. Computes arrays NCF and NCG which contain the same information as NUMF and NUMG but organized by participating sensor and participating parameter. Thus, for the ith complex,

NCF(n , I) = string of participating sensor pointers, packed 2 to a word, NCG(n , J, I) = string of derivative pointers for the Jth participating sensor, packed 6 to a word.

Furthermore, the arrays NUMCU and NUMCV are generated. Again, for the Ith complex:

NUMCU(n , İ) = string of pointers to access parameter library USC,
NUMCV(n , I) = string of pointers to access active parameter array.

Both of these strings are packed 6 to a word. Notice the reciprocal relationship between these strings: let the Jth element of the first be the number K— then the Kth element of the second string is the number J̇. Moreover, if a given entry, the Mth, in the second is zero, then the number M does not appear in the first string. These strings are used to transmit system parameters during differential correction in module GADSLS. See Figure 8.

5. Records the total number of participating sensors in the Ith complex, NSS(İ).

6. Records the maximum space that would be needed during outlier detection trials, NTS(I).

Control 8 - processes the constraints in a sensor complex:

1. Reads constraint names and multiplier codes.

2. Generates arrays NCHF and NCHG which are analogous to NCF and NCG. See Control 7, above.

3. Generates the array NUMCH which contains the string of constraint pointers.

4. Records the total number of constraints in the Ith complex, NHS(I).

Control 9 - not used.

Control 10 - not used.

Control 11 - loads perturbations for computing numerical derivatives. Blank fields are assumed to be implied perturbations as in Control 1. The array LINKUH is loaded with the intergers which specify the order (degree) of perturbation. See module GADSGC.

Control 12 - processes display controls:

1. Determines what is to be displayed.

2. Determines which complex is requested.

3. Determines the display level (quantity).

4. Determines which sensor outputs are requested in the display.

5. Determines what sensor-related data is requested. The various types of sensor-related plots are:

APLOT - raw data

FPLOT - predicted sensor output

DENSEF - same as above with interpolated functional values to create a continuous appearing curve. (See Figures 4 and 5 in Reference 1)

ENVELP - plot the one sigma envelope

TPLOT - plot the predicted telemetry values (calibrated theoretical output)

RPLOT - plot residuals

This information is transmitted to module GADSTV by utilizing control words KGRPHC($\dot{I}$). Let KGRAPH = KGRPHC($\dot{I}$). Then the Ith sensor complex will result in displays according to the bits appearing in KGRAPH. The meaning of the bits are as follows:

Power of 2 Bit	Significance if Bit is On.
0	Display 1st sensor in library.
1	Display 2nd sensor in library.
3	Display 3rd sensor in library.
.	.....
.	.....
.	.....
15	Display 16th sensor in library.
16	Not used.
17	Not used.
18	Display raw data (APLOT).
19	Display predicted points (FPLOT).
20	Generate intermediate points (DENSEF)
21	Generate ± one sigma envelope (ENVELP).
22	Display predicted telemetry (calibrated) points (TPLOT).
23	Display residuals (RPLOT).
24	Superimpose all sensor curves.
25	Not used.
26	Not used.
27	Plot final results of differential correction after all isolation of outliers trials, if any.
28	Plot final results after each differential correction loop converges; i.e., in-between each outlier detection trial, if any.
29	Plot the curves at each iteration of the differential correction loop.

6. Reads display controls. For details concerning these controls, see Reference 10, pages II-11, 12, 13.

Control 13 - processes request for numerical derivatives:

1. Reads names of parameters designated for numerical differentiation.

2. Tests for the presence of at least one complex, NC $\neq$ 0.

3. Tests for the presence of previously declared perturbation and related quantity (degree), NPERT $\neq$ 0. (See Control 11.)

4. Counts total number of perturbed calculations, NCUT.

5. Determines if the parameter was a calibration constant or a mounting constant. If so, flag bits are inserted into KSCUTS and KOCUTS, respectively. This is to indicate whether the perturbed attitude calculations can be skipped.

6. Generates the numerical derivative flags in KCCUTS. Thus, if the Jth system parameter was designated for numerical differentiation, then the 2^J bit in KCCUTS will be on. Moreover, if the said parameter was also a calibration or orientation (mounting) constant, the corresponding bit in KSCUTS or KOCUTS, respectively, will also be on.

Control 14 - processes auxiliary functions needed in a given sensor complex. Auxiliary functions include the state vector:

1. Loads auxiliary function names.

2. Generates the string of auxiliary module numbers LINKAU.

3. Counts the total number of auxiliary functions, NAUX. Each auxiliary module may generate several outputs. This is determined by NAUXDI. See module GADSDA.

4. Generates the switch variable KSAUX. The range and significance are:

1 = no auxiliary functions,

2 = integration of differential equations only,

3 = closed-form functions only,

4 = closed-form functions after integration,

5 = closed-form functions before integration.

Control 15 - processes differential equations to integrate in a given sensor complex. The state vector can be among them. The procedure is analogous to the previous one.

Control 16 - processes request to calibrate given sensors in a given sensor complex:

1. Reads sensor names.

2. Inserts calibration request flag into sensor output function string LINKF.

Control 17 - loads the list of parameter names which are sensor calibration constants and also loads the constant's pointer. The contents of the arrays CALIBP and KCPS are as follows:

CALIBP($\dot{I}$, 1) = a system parameter name appearing in array.

PARAMS. See Control 1.

CALIBP($\dot{I}$, 2) = a sensor name appearing in array OTITLE. See Control 5.

KCPS(I) = an integer pointer in the range 1, 2, 3, . . . 6, inclusive.

Prior to the return, module GADSIN interprets the preceding arrays. See below, Control 999.

Control 18 - loads the list of parameter names which are sensor orientation (mounting) constants and also loads the pointer. The action is analogous to the preceding.

Control 19 - processes special derivative codes. (Note that this action occurs under Control 5.)

Control 20 - illegal during regular processing. Refer to Control 420.

Control 21 - loads user identification for SC 4020 display. See page II-2, Reference 10.

Control 22 - loads problem definition. This is included for future use when non-Eulerian method of parametrization may be included in this module.

Controls 23-32 - illegal during regular processing. Refer to Controls 423-432.

The following controls are referenced only during the first call to GADSIN, when INRSET=1. During this phase, run cards are processed.

Control 401 - replaces the resident parameter names with those provided.

Control 402 - replaces the resident parameter estimates with those provided.

Control 403 - replaces the resident parameter accuracy with those provided.

Control 404 - replaces the resident parameter convergence requirements with those provided.

Controls 405-410 - illegal during initialization.

Control 411 - replaces the resident perturbations with those provided.

Controls 412-416 - illegal during initialization.

Control 417 - replaces resident parameter calibrations with those provided.

Control 418 - replaces resident parameter orientation (mounting) constants with those provided.

Control 419 - illegal during initialization.

Control 420 - loads physical card deck into RUNDEK, counts total number of cards, pads out array RUNDEK with STOP codes.

Control 421 - replaces the user resident identification with the one provided.

Control 422 - replaces problem definition with the one provided.

Control 423 - processes run deck modifications when card images are resident in array RUNDEK:

1. Clears N4, the card count excess.

2. Reads an UPDATE card containing I, J, K controls.

3. Determines whether there are any deletions. If not, J = 0.

4. Computes parameters needed in moving card images:

   N0 = +1 for moving card images backward,

   N0 = -1 for moving card images forward,

   N1 = 1 for referencing the start of a card image,

   N1 = 14 for referencing the end of a card image,

   N2 = the card image number source,

   N3 = the card image number destination,

   N = total number of computer words involved in the move.

To better understand these variables, refer to module MOVE.

5. Makes or deletes space using MOVE.

6. Reads K new card images.

7. Adjusts N4, the card count excess.

8. Continues with next UPDATE card after testing for STOP UPDATE.

<u>Control 424</u> - prints and counts the contents of RUNDEK.

<u>Control 425</u> - no action.

<u>Control 426</u> - starts initialization of GADS system.

<u>Control 427</u> - no action.

<u>Control 428</u> - starts initialization of GADS system.

<u>Control 429</u> - no action.

<u>Control 430, 1, 2</u> - no action.

<u>Control 999</u> - summary.

1. Uses LINKF as temporary work space to pack sensor complex data into NUMC1. Let LC denote a sensor complex, then the contents of NWORK2 are as follows:

NWORK2(LC, 1) = NUS(LC) = number of active parameters,
NWORK2(LC, 2) = NSS(LC) = number of active sensors,
NWORK2(LC, 3) = NTS(LC) = number of raw data points,
NWORK2(LC, 5) = NAUXS(LC) = number of auxiliary functions,
NWORK2(LC, 6) = NDEQS(LC) = number of differential equations,
NWORK2(LC, 7) = NCUTS(LC) = number of perturbed calculations,
NWORK2(LC, 8) = KSWAUX(LC) = KSAUX (see Controls 14, 15),
NWORK2(LC, 9) = NTRYS(LC) = NTRIES (outlier detection),
NWORK2(LC, 10) = IGNORX(LC) = ignore complex flag,
NWORK2(LC, 11) = IQUITS(LC) = number of iterations,
NWORK2(LC, 12) = MOTNS(LC) = type of motion,
NWORK2(LC, 13) = NHS(LC) = number of constraints.

2. Summarizes requests for adjusting calibration constants:

a) determines if the parameter name was actually included in PARAMS; if so, KU is the pointer,

b) determines if the sensor name was actually included in OTITLE; if so, JS is the pointer,

c) generates the control NUMCPJ and stores it in NUMCP(JS). Consider NUMCPJ as a string of six, six-bit numbers denoted by the symbols n1, n2, n3, ...n6. Then

n1 = system parameter pointer for raw calibration constant 1,

n2 = system parameter pointer for raw calibration constant 2,

....

n6 = system parameter pointer for raw calibration constant 6.

Note that a zero entry denotes no request. Hence, these codes are useful in accessing the refined calibration constant in the array of system parameters during differential correction. See modules of the type GADGO1.

3. Summarizes requests for adjusting orientation (mounting) constants. The action is analogous to the preceding. The variables NUMOP and NUMOPJ have similar roles as NUMCP and NUMCPJ, respectively.

4. Clears reset switch, INRSET. This variable is initially set to 1 in block data module GADSRD and is cleared here on the first call to GADSIN. When INRSET = 0, module GADSIN does not try to read cards but proceeds directly to the regular processing which starts at Control 135.

5.0 Usage: When module GADSIN is unable to complete the interpretation of the run deck, the alternate return is invoked. In this case, the user's main executive program should not call GADS.

Note that the GADS system has to be overlaid in computer core, module GADSIN must be called when GADS is called. The run deck is read only once. This means, however, that the common area RUNDEK must be retained in memory at all times. For further details, see Section 2.

Module GADSIX. (Initialize complex)

1.0 Calling Sequence: CALL GADSIX.

2.0 Category: FORTRAN subroutine, GADS executive, level 2.

3.0 Purpose: To initialize a sensor complex.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

EPS	Input convergence criteria. Normally set to .100E-05 in module GADSDA. Refer also to Run Deck, card G8-1.
EPS1	Output. Temporary storage for convergence criteria.
KCCUTS	Output. False position derivative flags. These flags affect GADSFC and GADSGC.
KOCUTS	Output. False position derivative flags associated with orientation (or mounting) constants.
KSCUTS	Output. False position derivative flags associated only with sensor calibration constants.
KU	System parameter pointer index.
LC	Input. Sensor complex pointer.
LINKAU	Output. Linkage array for auxiliary functions referenced in modules GADSAC and GADSDC.
LINKEJ	Output. Linkage array used in module GADFC1 to determine when module GADSAL can be skipped; i.e., when the coordinate transformation $\mathcal{E}^{g} = \mathcal{A}\mathcal{G}$ need not be recomputed.

LINKF	Output. Linkage array used in module GADSFC to determine the type of prediction function, the type of weighting, if any, and the type of calibration, if any.
LINKIC	Output. Linkage array used in modules of the type GADD01 to obtain initial conditions.
LINKRF	Output. Linkage array used in module GADSFC to control differentiation by the false position method.
LINKU	Output. Linkage array used in module GADSLS to control the transmission of system parameters.
LINKV	Output. Linkage array used in module GADSLS to control the transmission of system parameters.
LINKKG	Input. Type of parameter codes or indicators. Used in module GADSGC to determine special procedures during computation of derivatives. See Figure 3. Refer also to variable LNKKG1 under modules GADSGC and those of the type GADF01. These codes may take on the following values: 1 = pure parameter of the motion which does not affect sensor operand (see equation V.23, page 63, and page 90ff, Reference 1), 2 = parameter of the motion affecting sensor operand, 3 = parameter mounting constant, 4 = parameter calibration constant.
LNKKG1	Output. Linkage array used in module GADSGC to determine special procedures during the computation of the derivatives.

NCAUX	Input. The packed linkages LINKAU. Each sensor complex has its own LINKAU.
NCDEIC	Input. The packed linkages LINKIC.
NCF	Input. The packed linkages LINKF.
NFILLS	Output. Data fill counters to be cleared.
NS	Input. Number of sensors participating in this sensor complex.
NTOTF	Output. Total number of data fills counter to be cleared.
NU	Input. Total number of active system parameters in this complex.
PARAMS	Input. System parameter hollerith names.
POWERS	Input. Powers of 2 from 0 to 16, inclusive.
U	Output. Refer to this variable in module GADSLS.
UCUTS	Input. Perturbations used in module GADSFC for computing derivatives according to the method of false position.
UCUTS1	Output. Temporary locations for UCUTS.
USC	Input. Library of system parameters for all sensor complexes.
USREL	Input. System parameter-sensor relationship indicator, if any. See module GADSIN.
USREL1	Output. Temporary locations for USREL. Module GADSGC is the customer.
U1, U2	Output. Temporary locations for system parameters. See explanations under module GADSLS.

VNAMES Output. Temporary locations for active system parameter hollerith names.

5.0 Notes: This module provides the interface between module GADSLS and the resident controls which are maintained in packed form in named common areas. Its role is complementary to that of module GADSCS which is concerned with those aspects of initialization related to core work space allocation.

Module GADSLS. (Least-squares)

1.0 Calling Sequence: CALL GADSLS($, CTITLR).

2.0 Category: FORTRAN subroutine, GADS executive, level 1.

3.0 Purpose: To provide the executive decisions for finding the least-squares estimate of a set of system parameters.

4.0 Variables:

4.1 Explicit Inputs:

$	FORTRAN alternate return, in case of failure, to obtain convergence in differential correction procedure.
CTITLR	Hollerith identification of first requested sensor complex to be processed.

4.2 Explicit Outputs: None.

4.3 Important Interface Variables:

COMM — Input. This array contains the raw data and corresponding observation times. See Figure 5. Under user instructions, specific cells can be cleared during the procedure called detection of outliers as discussed in Note 3 below.

TORIGN(I) — Input. This array is important when ephemeris data and observation times are not referenced to the same base time or time origin. Its contents are:

TORIGN(1) = not assigned,
TORIGN(2) = day of year (or launch year),
TORIGN(3) = hour of day,
TORIGN(4) = minute of hour,
TORIGN(5) = millisecond of minute,

TORIGN(6) = year,

TORIGN(7 and 8) = double precision millisecond of year.

It is also necessary to provide these quantities for proper identification of printer output and displays. Refer also to modules of the type GADSSO.

USC(K, LC) Input and output. This array contains the initial estimates of system parameters. Upon successful processing of the LCth sensor complex, for example, USC(K, LC); K = 1, 2, ... 32 contains the system parameters as refined by the said complex. When a sensor complex fails to obtain convergence, the contents of USC(, LC) are undisturbed.

U(K, L) Output. Upon a normal return from GADSLS, this variable contains the most recently refined set of system parameters. Note that the refinement of system parameters cascades. That is, the successfully refined variables in a given sensor complex are transmitted to the ensuing complex as initial estimates. Hence, each complex benefits from the experience of its predecessors and the results in array U are cumulative. The user should not reference U(, 2).

LENGTH(J) Input. Number of raw data points for each sensor.

LOCATE(J) Input. Relative starting address in COMM for each raw data array.

LOCA Input. LOCA should always equal 1.

LOCT Input. Relative address in COMM of first observation time. See Figure 5.

4.4 Intermediate and Implicit I/O Variables:

ASTRSK	Hollerith asterisk. Printed adjacent to names of system parameters when they have been refined.
COMM	Common area containing raw data, sample times, and working space.
CRGAMM	Critical gamma. A criterion to decide whether to shorten the step size in the method of steepest descent. See Note 1 and Reference 2. The critical value of gamma is arbitrarily set at 45^{o}. See Figure 7.
CTITLE	Hollerith titles used for identifying sensor complexes.
CTITLN	Hollerith title of next sensor complex to be processed.
CTITLS	Hollerith data related to sensor complexes. This array is EQUIVALENT to CTITLE. Also see Run Deck, card G14a-1.
CTITLX	Hollerith title of alternate next sensor complex in cast of error (divergence in differential correction).
CTITL1	Hollerith title of current sensor complex.
CTOLER	Criterion, in standard deviations, for detecting an outlier. Refer to Note 3 and card G14a-1.
DEGF	Degrees of statistical freedom. See NDEGF.
DIAGN	Hollerith diagnostics.
DONE	Logical flag used to indicate state of convergence.
EPS	Convergence criteria. Refer also to Run Deck, card G8-1.
EPS1	Temporary storage for convergence criteria.
IERROR	Flag used to indicate if error return should be used.

IFRESH	Fresh outlier counter. See Note 3.
IGNORC	Flag used to indicate if a sensor complex should be ignored. Refer to Run Deck, card G14a-1.
IGOOD	Useful data point counter.
ITRIAL	Temporary variable used in detection of outliers. See Note 3.
IPRCNT	Criterion used in detection of outliers. See Note 3 and Run Deck, card G14a-1.
IQUIT	Maximum number of iterations.
JFRESH	Fresh outlier counter for previous trial. See IFRESH.
JTIME	This variable is defined so that if COMM(I) is a specific observed data, then COMM(I + JTIME) is the associated observation time in milliseconds. Refer to Section 2. Thus JTIME + LOCT - LOCA.
KGRAPH	Flag indicating request for graphs.
LAMBDA	Pointer index used to distinguish between gradient method (steepest descent) and Newton or Taylor method: LAMBDA = 1: gradient method, LAMBDA = 2: Newton method.
LAMBDM	Maximum LAMBDA. Normally LAMBDM=2; when the gradient method is no longer needed, LAMBDM=1.
LAMBIN	Initial value for LAMBDM.
LC	Index for active sensor complex.
LINKAU	Linkage array. See NCAUX and documentation modules GADSAC, GADSDC.
LINKF	Linkage array. See documentation GADSFC.

LINKG	Linkage array. See documentation GADSGC.
LINKU	Linkage to control the transmission of parameters from their standard positions (as in U, PARAMS) to their temporary positions in the normal equations (as in U1, U2, DV, VNAMES). Thus, if LINKU(KU)=K, the active parameter K being refined by differential correction is the KU system parameter. See variable LINKV.
LINKV	Linkage to control the transmission of parameters from their temporary locations to their standard positions. Thus if LINKV(K)=KU, the K system parameter is being refined by differential correction.
LINKKG1	This array contains the same data as LINKKG but arranged as dictated by LINKU so that the elements of this array correspond to those of U1, U2, DV, and VNAMES.
LOCA	Relative location of the first cell of the raw data input in COMM.
LOCAIN	Relative location of the first cell of the inverted matrix of coefficients. See documentation, module GADSCS and Figure 10.
LOCAMA	Relative location of the first cell of the matrix of coefficients. See module GADSCS and Figure 10.
LOCANR	Relative location of the normalized matrix of coefficients. See module GADSCS and Figure 10.
LOCAXY	Auxiliary for LOCAMA.
LOCDU	Relative location of the differential corrections (double precision).

LOCD1	Relative location of the residuals. See module GADSCS.
LOCEND	Relative location of the end of the work space, COMM.
LOCFLG	Relative location of the flags used in tagging outliers. See module GADSFC and Figure 10.
LOCT	Relative location of the first cell of the time array.
LOCTEL	Relative location of the first cell of the predicted telemetry array. At this writing, LOCTEL=LOCF1. See Figure 10.
LOCTOP	Relative location of the highest cell used for work space not including those that may be required by module GADSTV to generate dense plots. Module GADSTV tests for availability of space. See Figure 10.
LOCYMA	Relative location of the first cell of the gradient vector. See Figure 10.
LOCYNR	Relative location of the first cell of the normalized gradient vector. See module GADSMN and Figure 10.
LOCY1	Relative location for a series of arrays computed in module GADSMN during differential correction. See Figure 10.
NAUX	Total number of active auxiliary functions.
NC	Total number of active complexes.
NCALLS	Number of calls counter.
NCAUX	Packed linkages LINKAU. Each complex has its own linkage.
NCF	Storage array for the packed linkages LINKF. See modules GADSIN and GADSFC.

NCG	Storage array for the packed linkages LINKG. See modules GADSIN, GADSGC.
NCOMM NDEBUG	EQUIVALENT to COMM.
NFILLS	Fill-data counters for various sensors: NFILLS(1) = number of fills for sensor 1, NFILLS(2) = number of fills for sensor 2, etc.
NONCE1	A flag, initially set to 1 and later set to 0 in module GADSAL, which serves to prevent unnecessary re-initializing.
NONCE2	Similar to NONCE1. Prevents unnecessary calculations in module GADA01.
NONCE4	Similar to NONCW1.
NS	Total number of participating sensors in a given complex.
NT	Total number of observed data points for a given sensor.
NTOTF	Counter for total number of fill-data.
NTOTR	Counter for total number of rejected data points (outliers).
NTRIAL	A counter initially set to zero for each complex. NTRIAL is incremented by 1 at the conclusion of each successful attempt to isolate outliers. See Note 3.
NTRIES	Maximum NTRIAL. See Run Deck, card G14a-1, and module GADSIN.
NU	The total number of participating parameters in a given complex.
NUMC	Storage array for packed flags.

NUMCU	Storage array for packed linkages LINKU.
NUMCV	Stroage array for packed linkages LINKV.
NV	Total number of active variables in a complex, inclusive of constraints. Note that constraints are not used at this writing.
PARAMS	Permanent storage for system parameter names (hollerith).
POID1	Printout identification (hollerith).
POID2	Printout identification (hollerith).
POID3	Printout identification (hollerith).
POWERS	Permanent array containing the powers of 2 up to 16, inclusive, used in Boolean-like decision processes.
PROGID	Hollerith program identification.
RSUMSQ	Square root of sum of squared error: RSUMSQ(1) is used by gradient method, RSUMSQ(2) is used by Taylor method.
SDEV	Standard devision of fit.
SUMSQ	Sum of squared errors.
TEMP1-8	Eight temporary work locations.
TFIXED	Time origin in millisecond of year (double precision).
TIME	Time of a specific data. (A sample time.)
TLABEL	Hollerith labels for the time printout.
TOLENV	Criterion tolerance envelope, in standard deviations, for isolating outliers.

TORIGN	Array containing the time origin in days, hours, minutes, milliseconds, millisecond of day, and millisecond of year. See coding of module GADSSO, EQUIVALENCE statements, and Section 2.
U	Transient storage for the system parameters. During differential correction, U always contains the best results from the previous iteration and is a back-up set of system parameters. Note, the symbol U in all other programs coincides with UTEMP, not with U.
UACC	Array of numbers indicating the accuracy with which the system parameters are known: 1.0 = perfect accuracy, 0.0 = no knowledge at all.
UCUTS	Array of small numbers of perturbations for use in numerical differentiation. See module GADSFC.
UCUTS1	Same as UCUTS but arranged according to LINKU.
UEST	Array of system parameter estimates.
USC	Array of system parameters as refined by each sensor complex. Thus, USC(KU, LC) is the KU parameter as refined by the LC sensor complex.
UTEMP	Temporary locations for system parameters.
U1	EQUIVALENT to U1A.
U1A	Transient storage for the system parameters while making differential correction trials. Thus, U1A(K, 1) is the trial for gradient method, U1A(K, 2) is the trial for Newton method.

U2	EQUIVALENT to U1A(1, 2).
VNAMES	Hollerith names for the variables in array V.
XKDU	Multiplying factor for the gradient method step size. Refer to Note 1, symbol k.
XKDUCR	When XKDU<XKDUCR, the gradient has vanished.
XLAMB	Donald Marquardt's λ. See Reference 2 and Note 1, symbol x λ.
XLAMBC	Critical value of XLAMB. When XLAMB is greater than XLAMBC, convergence is not accepted regardless of the magnitudes of differential corrections. See module GADSDA.
XLAMBM	Maximum legitimate value for XLAMB. Should the latter variable become excessive, differential correction is not assumed divergent. See Diagnostic Printout #6 and module GADSDA.
XLAMIN	Initial value of XLAMB. The user may wish to modify this quantity. The general rules are as follows: a) the more accurate the initial estimates of the parameters are known, the smaller XLAMIN, b) the more linear the least squares problem, the smaller XLAMIN.

5.0 Notes:

(1) Method of Maximum Neighborhood:

Module GADSLS uses Marquardt's method of maximum neighborhood reported in Reference 2. This is a technique for least-squares differential correction which combines the best qualities of the Newton or Taylor method and the method of steepest descent (gradient method).

The first important step in applying Marquardt's technique is to normalize the Taylor equations of condition, equations III. 24, page 22, Reference 1.

$$N \Delta U = \Gamma.$$

Reference 2 discusses the reasons for normalizing these equations. Normalization is achieved as follows:

$$A_{ij} \equiv N_{ij} / (N_{ii} \; N_{jj})^{1/2}, \tag{2}$$

$$\delta_i^* \equiv \Delta U_i \; N_{ii}^{1/2}, \quad \text{and} \tag{3}$$

$$g_i \equiv \Gamma_i / N_{ii}^{1/2}. \tag{4}$$

Combining (2), (3), (4) with (1), we have

$$A \; \delta^* = g. \tag{5}$$

As stated in Reference 2, this choice of scale amounts to using the standard deviations of the derivatives $\partial \tau / \partial \upsilon$ as units of measure and allows us to regard A as a matrix of correlations.

After normalization, the next step is to construct the equation

$$(A + X_\lambda I) \; \delta = g. \tag{6}$$

Clearly, as X_λ approaches zero, this equation becomes identical with the Taylor method; i.e., equation (5). Thus,

$$\lim_{X_\lambda \to 0} \delta = \delta^*, \tag{7}$$

and, hence, is sometimes called a quasi-Taylor vector. On the other hand,

$$\lim_{X_\lambda \to \infty} \delta = g/X_\lambda. \tag{8}$$

In this case, δ becomes parallel to the gradient vector g. The limits (7) and (8) represent the pure Taylor and pure gradient methods, respectively. The latter method guarantees minimization of the squared error but converges slowly. The former may diverge in nonlinear problems but has good convergence properties. Marquardt's method consists of adjusting the mixture parameter x_λ to obtain optimum convergence characteristics. The program symbol for x_λ is XLAMB.

Marquardt's strategy for convergence is illustrated in Figure 7. Several quantities not yet described are involved. The first is the root-mean-squared error σ. The corresponding program symbol is SDEV. This is the unbiased standard deviation of the residuals (or of the fit) and is obtained from

$$\sigma = (q/(N - NU))^{\frac{1}{2}}, \tag{9}$$

where q is the sum of squared errors (residuals), N is the total number of data points, and NU is the number of independent parameters. Another important quantity is γ:

$$\gamma \equiv \arccos\ (\delta \cdot g / |\delta|\ |g|), \tag{12}$$

the angle between the quasi-Taylor vector δ and the gradient vector g. AGAMMA is the program symbol for γ. In linear problems these vectors are parallel. However, in non-linear problems they are not, and hence, γ can be considered a measure of non-linearity. In Marquardt's method, this angle is monitored in order to determine when the mixture parameter x_λ should be increased; i.e., when more gradient and less Taylor is needed. For this purpose, γ_c, the critical angle, arbitrarily set at 45^o, is used. Still another important quantity is k for which the corresponding program symbol is XKDU. In order to understand the purpose of k, assume that U^n be the array $\{u_1, u_2, \ldots\}^n$ of system parameters obtained at the nth iteration step. Thus,

$$u_i^{n+1} = u_i^n - \Delta u_i\ . \tag{10}$$

Then k is defined by

$$\Delta u_i \equiv k\, \delta_i\, N_{ii}^{\ 1/2}. \tag{11}$$

Evidently k is a parameter to control the step size ΔU and is normally set to 1.0 but may be halved repeatedly when difficulty is found in reducing σ. In such a situation, only the gradient solution is needed.

For further clarification of Marquardt's technique, refer to Figure 7 and Reference 2.

(2) <u>The Transmission of System Parameters</u>:

The application of the method described above results in the need for maintaining several sets of system parameters which are described in the following paragraphs. (See Figure 8.)

The array USC(KU, LC); $1 \leq KU \leq NU$, $1 \leq LC \leq NC$, contains NC sets of NU initial values for the system parameters. (One parameter set is assigned to each sensor complex.) During the processing of each sensor complex, the LC set of parameters is transferred to the array U where it will be updated only after each successful iteration of the differential correction loop. U is thought of as a back-up array of parameters. The array U1A(KU, LAMBDA); LAMBDA = 1, 2 on the other hand, contains the sets of trial system parameters used in the strategy for convergence by the maximum neighborhood method. Initially, U1A is identical with U, but U1A is modified at every trial differential correction step. In referring to the program, note that U1 is equivalent to U1A(1, 1), U2 to U1A(1, 2).

In differential correction, therefore, three possibilities arise:

a) U1 may be the best choice,

b) U2 may be the best choice,

c) Both U1 and U2 produce worse results than U.

Clearly, the corresponding actions are

a) Transfer U1 to U,

b) Transfer U2 to U,

c) No transfer, respectively.

Because all worker programs make strict reference to UTEMP for the system parameters, this array must always contain the intended set of system parameters. The transmission of system parameters is illustrated in Figure 8. Note that this method is more efficient, in terms of computer time and core space, than the use of calling sequences. In GADS, the unusually large number of worker subroutines makes calling sequences impractical.

(3) Detection of Outliers:

If the user desires, GADSLS will attempt to detect and discard outliers. These are defined as observed data points which fall outside a certain neighborhood centered at the predicted theoretical output. This neighborhood is measured in units of σ (standard deviations) and is given by the user. Refer to Run Deck, card 14a-1.

After convergence has been achieved, if the user has requested the detection of outliers (i.e., if he has punched non-zero in the field corresponding to NTRIES in the said card), the fitting process will be repeated neglecting outliers. The repetition of a fitting process is called a trial and the number of trials is counted by the variable NTRIAL. GADSLS will repeat the trials until NTRIAL = NTRIES(LC). Figure 9 illustrates the strategy for detection and isolation of outliers. Note that during each trial, the number of fresh outliers is counted in variable IFRESH. When this counter increases from one trial to the next, the isolation of outliers is considered out of control and halted.

This method for isolation of outliers is crude and should be avoided. When the data contain random errors with a definite well-behaved statistical distribution, this method is, in fact, erroneous. It may be useful, however, in detecting a wild point such as may result from bit dropouts. Bit dropouts may occur in digitizers, transmitters, receivers, and in processing magnetic tapes. Sometimes wild points may be caused by bit dropins which are products of electronic crosstalk (observed in OSO-B). Wild readings caused by geometric shadowing can, and should, be handled by the geometric operators. See Reference 1 and refer to modules of the type GADF01, 02, 03....

Module GADSML. (Main data loop)

1.0 Calling Sequence: CALL GADSML(A, Y, NL).

2.0 Category: FORTRAN subroutine, GADS executive, level 2.

3.0 Purpose: To perform the calculation of all required predicted functions, derivatives, and residuals for a given sensor complex. These calculations are performed for each pertinent observation sample time.

4.0 Variables:

4.1 Explicit Inputs:

NL	Total number of active system parameters.

4.2 Explicit Outputs:

A(NL, NL)	Unnormalized matrix of coefficients in the least-squares equations of condition. At this writing, A is in double precision. See equations III. 22a, b, c, and III. 25a, Reference 1.
Y	Gradient vector in the least-squares equations of condition. At this writing, Y is also in double precision. See equations III. 22d and III. 25b, Reference 1.

4.3 Intermediate and Implicit I/O Variables:

ADMH	Input. Step size for numerical integration module. Refer to module ADAMS.
ADMT	Current time (milliseconds) in the said numerical integration module.
ATEMP	Temporary location for observed data.
DEL	Computed residual; i.e., the difference between predicted and observed functions, DEL $= \tau_i - y_i$. See equation III. 11, Reference 1.

IGNORE	A flag. When a given data point is found objectionable for any of several reasons, IGNORE flag is set to 1. Otherwise it is zero. See module GADSFC.
IGOOD	Counter for acceptable data points.
IPRECI	Internal FORTRAN parameter. IPRECI is equal to 1 or 2, depending on whether A and Y (output variables) are single or double precision.
IT	Pointer index for the i data sample.
JETRAN	Flag to inhibit all calculations except $\mathcal{E} = \mathcal{A}\mathcal{L}$ transformation. Refer to module GADSFC.
JS	Pointer index for the j sensor.
JTIME	See JTIME under module GADSCS.
LAUXG	Starting relative location of the auxiliary functions. See modules GADSLS, GADSCS.
LDEQE	Starting relative location of the E array for the numerical integration module ADMSET.
LDEQF	Starting relative location of the F array for the numerical integration module ADMSET.
LDEQG	Starting relative location of the G array for the numerical integration module ADMSET.
LDEQP	Starting relative location of the P array for the numerical integration module ADMSET.
LDEQZ	Starting relative location of the Z array for the numerical integration module ADMSET.
LDEQ1	Total number of differential equations, including perturbed equations for use in numerical differentiation, to be integrated under module ADAMS.

LOCATE	Array containing relative starting addresses of the raw input data for each sensor. Refer also to module GADSIN, GADSLS, and Figure 5.
LOCFLG	Starting address of the flag array used in outlier detection. See modules GADSLS and GADSFC.
NAUX	Total number of auxiliary functions. See Run Deck, card G14b.
NCUT	Total number of parameters for which numerical differentiation (regula falsi) has been requested. See Run Deck, card G14c.
NDEGF	Total number of observations reduced by the to number of outliers, fills, and active parameters.
NDEQ	Number of unique differential equations being integrated under module ADAMS. Note that $NDEQ1 = NDEQ * NCUT.$
NDEQ1	Total number of differential equations being integrated including perturbed equations used to estimate derivatives by the method of false position. See variable NDEQ.
NONCE1	See module GADSLS.
NONCE2	See module GADSLS.
NONCE3	See module GADSLS.
NONCE4	See module GADSLS.
NS	Number of active sensors in a given sensor complex.
NT	Number of observations for a given sensor.
NTOTF	Total number of fill-data.

NTOTR	Total number of outliers or rejects.
NU	Number of active system parameters.
SUMSQ	Sum of squared weighted residuals.
U2	Array of partial derivatives. See module GADSGC.

Module GADSMN. (Maximum neighborhood)

1.0 Calling Sequence: CALL GADSMN($, A, B, Y, DU, ANORM, V, DV, XLAMBD, NL).

2.0 Category: FORTRAN subroutine, GADS executive, level 2.

3.0 Purpose: To compute the quasi-Taylor or differential correction vector.

4.0 Variables:

4.1 Explicit Inputs:

$	Error return in case of singular covariance matrix; i.e., matrix of coefficients.
B(NL, NL)	Double precision normalized covariance matrix.
Y(NL)	Double precision normalized gradient vector.
ANORM(NL)	Double precision normalization factors.
V(NL)	Array of intended system parameters.
XLAMBD	Marquardt's mixture parameter x_λ. See Note 1, Module GADSLS.
NL	Total number of active parameters.

4.2 Explicit Outputs:

A(NL, NL)	Double precision inverse of the quasi-Taylor matrix of coefficients.
DU(NL)	Double precision quasi-Taylor solution or gradient vector.
DV(NL)	Single precision equivalent of DU.
V(NL)	Single precision corrected system parameters.

4.3 <u>Intermediate and Implicit I/O Variables:</u>

DPR	Degrees per radian.
SUM1	$\lvert \delta * \rvert^2$. See Note 1, module GADSLS.
SUM2	$\lvert g \rvert^2$. See Note 1, module GADSLS.
SUM3	$\delta * \cdot g$. See Note 1, module GADSLS.
TEMP3	$\lvert \delta * \rvert$. See Note 1, module GADSLS.
TEMP4	$\lvert g \rvert$. See Note 1, module GADSLS.
TEMP7	$\delta * \cdot g / \lvert \delta * \rvert \lvert g \rvert = \cos(\gamma)$. See Note 1, module GADSLS.
TEMP8	γ. See Note 1, module GADSLS.
XKDU	k. See Note 1, module GADSLS.

Module GADSNE. (Normal equations)

1.0 Calling Sequence: CALL GADSNE($,A,Y,NK).

2.0 Category: FORTRAN subroutine, GADS executive, level 2.

3.0 Purpose: To normalize the equations of condition.

4.0 Variables:

4.1 Explicit Inputs:

$	Alternate return in case of division by zero.
A(NK,NK,1)	Double precision matrix of coefficients as computed in GADSML.
Y(NK,1)	Double precision gradient vector as computed in GADSML.
NK	Total number of active variables.

4.2 Explicit Outputs:

A(NK,NK,2)	Double precision normalized matrix of coefficients.
Y(NK,2)	Double precision normalized gradient vector.
Y(NK,5)	Double precision normalization factors.

4.3 Intermediate and Implicit I/O Variables:

KSOS	Diagnostic print selector. See Section 4.

5.0 Notes:

5.1 The Normal Equations: Calculating the elements of the normal equations involves the use of equations III. 22, 23, and 24 in Reference 1. This reference uses the symbols N and Γ in place of A and Y, respectively. Γ (or Y) is called the gradient vector (see Reference 2) because it is closely related to the gradient of the squared error function q. Also see equation III. 10, Reference 1.

5.2 <u>Normalization:</u>

The procedure for normalizing the matrices A and Y is described in Note 1, module GADSLS.

Module GADSPT. (Parameter transfer)

1.0 Calling Sequence: CALL GADSPT.

2.0 Category: FORTRAN subroutine, GADS executive, level 3.

3.0 Purpose: To transfer the refined parameters from one sensor complex to the next executable sensor complex.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

JS	Sensor pointer index.
LC	Input. Next executable sensor complex pointer index.
LINKCP(J)	Input. These quantities are defined in module GADSSP.
LINKF(J)	Input. See GADSFC and GADSIN.
LINKOP(J)	Input. See GADSSP.
NS	Input. Total number active sensors in sensor complex LC.
NUMCP(J)	Input. Refer to GADSSP.
NUMOP(J)	Input. Refer to GADSSP.
N0	Index to select a certain sensor in the sensor library.
OBCOEF	Input. Library of observable (sensor) calibration coefficients. See Run Deck, card G13-8.
OBMOUN	Input. Library of observable orientation (mounting) constants. See Run Deck, card G13-7.
U	Input. The refined system parameters. U corresponds to UTEMP in module GADSLS. See Note 2, GADSLS.

USC Output. The library of system parameters. (See the discussion referenced under variable U.)

5.0 Notes:

5.1 Source and Destination Addresses: Since the purpose is to transmit the refined (improved) parameters from the incumbent sensor complex to its successor, it is clear that the source is UTEMP, the most refined set of system parameters in GADSLS. The destination is simply USC, the library of sensor complex parameters.

5.2 Geometric and Transfer Function Operators: When geometric and/or transfer function operator parameters are being refined, the results should be returned to their respective libraries, namely OBMOUN and OBCOEF.

Whether the N0 sensor has such parameters is determined by variables NUMOP and NUMCP. These are computed in GADSIN.

Module GADSR1. (Sensor-related constants, type 1)

1.0 Calling Sequence: CALL SENSR1.

2.0 Category: CALL SENSR1.

3.0 Purpose: To compute cartesian direction cosines of a vector defined by two angles Φ and Θ.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

KGRADK	Input. KGRADK indicates a request for gradient of geometric operator: 0 – no request, 1 – request gradient.
OGAMMA	Output. Fast access area for cartesian direction cosines of observable (sensor) sensitive axis.
OGAMMD	Output. Fast access area for the derivatives of the direction cosines with respect to the raw geometric constants.
OMOUNT	Input. Raw geometric constants in fast access area.
TEMP1	Φ, radians. See Run Deck, card G13-7.
TEMP2	Θ, radians. See Run Deck, card G13-7.
TEMP3	cos Θ.
TEMP4	sin Θ.
TEMP5	cos Φ.
TEMP6	sin Θ.

5.0 Notes:

5.1 Purpose:

This module is designed for sensors whose outputs can be thought of as a dot product between the sensor's sensitive axis and some environmental variable such as the magnetic field or solar radiation. Hence, the raw geometric constants are interpreted as Φ and Θ. See Figure 16.

When the user wishes to account for a different type of sensor, he should observe that:

1. The outputs computed are used by modules of the type GADF01, GADG01, and GADC01. This means that he is free to use his own output area instead of OGAMM and OGAMMD, if he desires, provided that the said customer modules are programmed accordingly.

2. The raw constants appear in OMOUNT just as they were read by module GADSIN, card G13-7, and are referenced only by modules of the present type. Therefore, the user is free to define these constants in whatever way is convenient.

As stated under the Notes for module GADSSR, the main concern of these modules is to reduce the amount of computation occurring inside the main loop.

Note that this module should make preparation for computing the operators $\Box\,\mathscr{K}$ and $\nabla\,\mathscr{K}$, if necessary. These are the derivatives with respect to system parameters affecting $\mathscr{K}$.

5.2 Derivatives:

For example, consider a geometric operator such as considered above in 5.1. The ideal sensor output is

$$f = \quad S' = K \cdot S' \qquad (1)$$

See Reference 1, equation II. 7. S' represents the environmental variable S expressed in body coordinates. Rewriting (1) in detail using direction cosines:

where $$f = \sum_{i=1}^{3} \gamma_i S_i^1 , \tag{2}$$

$$\gamma_1 = \cos\theta \cos\Phi \tag{3a}$$

$$\gamma_2 = \cos\theta \sin\Phi \tag{3b}$$

$$\gamma_3 = \sin\theta. \tag{4}$$

Since the constants γ_i are simple multiplicative factors, $\Box \mathscr{H} = \mathscr{H}$

On the other hand, $$\nabla \mathscr{H} = \left\{ 0, 0 \ldots, \frac{\partial \mathscr{H}}{\partial \phi}, \frac{\partial \mathscr{H}}{\partial \theta}, \ 0, 0 \ldots \right\} \tag{5}$$

The non-vanishing derivatives are:

$$\frac{\partial}{\partial \Phi} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} = \begin{bmatrix} -\cos\theta \sin\Phi \\ \cos\theta \cos\Phi \\ 0 \end{bmatrix} \tag{6}$$

$$\frac{\partial}{\partial \theta} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} = \begin{bmatrix} -\sin\Theta \cos\Phi \\ -\sin\Theta \sin\Phi \\ \cos\Theta \end{bmatrix} \tag{7}$$

Module GADSSR. (Sensor-related calculations)

1.0 Calling Sequence: CALL GADSSR.

2.0 Category: FORTRAN subroutine, GADS executive, level 3.

3.0 Purpose: To calculate quantities used in the geometric operator starting from the raw mounting constants.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None

4.3 Intermediate and Implicit I/O Variables:

N6	Input. Type of geometry: N6=1 - fixed vector mounting, N6=2 - fixed axis, N6=3 - special, N6=4 - spare. Also see module GADSIN, variable SCODE4 and Run Deck, card G13-7.
OBGAMM	Output. Library of observable sensor-related mounting constants. See variable OBMOUN in module GADSIN and Run Deck, card G13-7.
OGAMMA	Output. Fast access area for direction cosines.
OMOUNT	Output. Fast access area for raw mounting constants.

5.0 Notes:

5.1 Purpose:

This module is used to generate the terminal geometric constants needed during computation of the predicted sensor outputs. That is, this module prepares the quantities needed in applying geometry considerations: The meaning of the term terminal constants is that all possible preprocessing is done. This reduces the amount of time consumed during the highly repetitive operations occurring within the main loop. (See module GADSML.) Note, for example, that module GADSR1 computes the cartesian direction cosines of the sensor axis starting from the two angles, ϕ and θ , which describe the said axis in body coordinates. Similarly, the user may code his own geometric operator preprocessing in modules GADSR2, 3, 4. In all cases, however, the main idea is to do as much as possible before modules of the type GADF01 are called.

5.2 Derivatives of the Geometric Operator:

In the event that certain raw mounting constants are to be adjusted by differential correction, the modules GADSR1, 2, . . . should also prepare the derivatives of the geometric operator. Refer to module GADSR1.

Module GADSSE. (Solar ephemeris)

1.0 Calling Sequence: CALL GADSSE.

2.0 Category: FORTRAN subroutine, GADS worker.

3.0 Purpose: To compute the cartesian components of the sun line-of-sight unit vector.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

DCOSF1	Double precision cos (FI).
DM	Double precision mean daily motion.
DSINE	Double precision sin (E).
DSINFI	Double precision sin (FI).
E	Double precision eccentric anomaly.
EC	Double precision orbital eccentricity.
EC1	Double precision table of EC for years 1965 through 1969.
FI	Double precision mean obliquity of YEAR.
FI1	Double precision table of FI for same years.
FL	Double precision longitude referred to mean equinox of YEAR.
FL1	Same as FL.
FLP	Double precision mean longitude of perigee referred to mean equinox of YEAR.

FLP1	Double precision table of FLP for same years.
MO	Double precision mean anomaly at January 0, YEAR.
MO1	Double precision table of MO for same years.
MSMM	Double precision mean motion, degrees per millisecond.
NCALLS	Number of calls counter.
ORBVEC	Output. Orbit-related array. The first column of ORBVEC is reserved for the sun vector which is the output of this module.
RADC	Double precision conversion (from degrees) to radians.
SFL	Double precision sin (FL).
TFIXED	Input. Double precision millisecond of year time origin for current satellite pass.
TIME	Input. Elapsed time since TFIXED for which calculations are desired.
TIMED	Double precision millisecond of year for which calculations are desired.
TORIGN	Input. Array of quantities defining the time origin of the current satellite pass. See EQUIVALENCE statements and module GADSSO.
YEAR	Input. Year for which calculations are desired: $1965 \leq YEAR \leq 1969$.

5.0 Notes:

5.1 Mean Anomaly:

By definition the mean anomaly is $M = M_0 + \left(\frac{dM}{dt}\right) \Delta t$ where dM/dt is the mean motion.

5.2 Eccentric Anomaly:

Neglecting fourth and higher order terms, the eccentric anomaly E may be expanded in terms of M and the eccentricity e as follows:

$$E = M + \frac{e \sin M}{1 - e \cos M} - \frac{1}{2}\left(\frac{e \sin M}{1 - e \cos M}\right)^3.$$

See page 134, Reference 17.

The accurate equation $E = M + e \sin E$ is not used because it is transcendental.

5.3 Longitude Referred to Mean Equinox of YEAR:

If λ_0 is the sun's longitude of perigee referred to the mean equinox of YEAR, then the instantaneous longitude, λ, referred to the same origin is $\lambda = \lambda_0 + E$. See Figure 18.

5.4 Cartesian Components:

From Figure 18 it is evident that the sun's cartesian components in the G. E. I. system are:

$$x = \cos \lambda$$
$$y = \sin \lambda \cdot \cos i$$
$$z = \sin \lambda \cdot \sin i$$

where i is the mean obliquity (or inclination). For further details see References 13 and 17.

5.5 Interface Considerations:

The calling program must provide the following quantities: (a) TFIXED, (b) TIME, (c) YEAR. Then the module GADSSE will compute the normalized components x, y, z of the sun's line-of-sight vector in the G. E. I. system. This output is placed in ORBVEC as follows:

$$ORBVEC(1,1) = x$$

$$ORBVEC(2,1) = y$$

$$ORBVEC(3,1) = z.$$

5.6 Verification of Results:

When comparing these outputs with those of Reference 14, the user is reminded that the latter are given in astronomical units (A.U.). That is, they are not normalized.

Module GADSSO. (Sensor operand)

1.0 Calling Sequence: CALL GADSSO.

2.0 Category: FORTRAN subroutine, GADS worker.

3.0 Purpose: To generate the sensor operand and its derivatives, if necessary.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

BALF	Output. Right ascension of magnetic field, radians.
BALFDG	Same as BALF in degrees.
BCAP	Output. Magnetic field, in inertial coordinates.
BDEL	Output. Declination of magnetic field, radians.
BDELDG	Same as BDEL in degrees.
BTIME	Same as TIME.
BTOT	Output. Magnitude of magnetic field.
COAB	Output. Cosine of BALF.
COAS	Output. Cosine of SALF.
CODB	Output. Cosine of BDEL.
CODS	Output. Cosine of SDEL.
DPR	Degrees per radian.
DP1, 2	Double precision variables.
DTIME	Double precision time, millisecond of year.
ECLIPS	Output. Integer solar eclipse indicator.

IBM	Input. IBM tape bit structure indicator.
J	Tape rewind counter.
K	Output. Rewind orbit tape flag: $K \geq 0$: no rewind, $K = -1$: rewind.
KBALF	Address of BALF in tape input record.
KBDEL	Address of BDEL in tape input record.
KBTOT	Address of BTOT in tape input record.
KBX, Y, Z	Address of BX, BY, BZ in tape input record.
KDAYOR	Address of day in tape input record.
KECLPS	Address of eclipse in tape input record.
KHOURO	Address of hour in tape input record.
KORBNO	Address of orbit number in tape input record.
KOTAPE	Input. Orbit tape unit.
KSALF	Address of SALF in tape input record.
KSDEL	Address of SDEL in tape input record.
KSTOT	Address of STOT in tape input record.
KSX, Y, Z	Address of SX, SY, SZ in tape input record.
KWORDS	Input. Number of words in tape input record.
KYEARO	Address of year in tape input record.
MSDORB	Address of millisecond of day in tape input record.
NITEMS	Input. Number of items per data set.
NSETS	Input. Number of item sets in one tape input record. Note that one item set is a group of words defining a complete orbital data point.

ORBIT	Input area.
ORBVEC	Output. Sensor operands. This particular module places the solar line-of-sight unit vector and the magnetic field vector in ORBVEC(I, 1); I = 1, 2, 3 and ORBVEC(I, 2); I = 1, 2, 3, respectively.
RPD	Radians per degree.
SALF	Output. Right ascension of solar line-of-sight vector, radians.
SALFDG	Same as SALF in degrees.
SCAP	Output. Solar unit line-of-sight vector.
SDEL	Output. Declination of solar line-of-sight vector, radians.
SDELDG	Same as SDEL in degrees.
SIAB	Output. Sine of BALF.
SIAS	Output. Sine of SALF.
SIDB	Output. Sine of BDEL.
SIDS	Output. Sine of SDEL.
STOT	Output. Distance to sun. Not computed at this writing.
TFIXED	Input. Double precision time origin, millisecond of year.
THOUR	Input. Time origin hour of day.
TIME	Input. Time in milliseconds since time origin for which output is desired.
TMILLS	Input. Time origin, millisecond of minute.

TMINUT	Input. Time origin, minute of hour.
TORIGN	Input. Array of numbers defining time origin. Note EQUIVALENCE statements.
TYEAR	Input. Time origin, day of year (or of launch year).
XMSDAY	Milliseconds per day, double precision.
XMSYR	Milliseconds per year, double precision. (Note GADS own system common area provides the same quantity, namely YMILLS.)

5.0 Explanation:

5.1 Main Considerations:

This module should calculate the sensor operands; i.e., the environmental variables upon which the sensing devices act. See 5.2, Specialization and Modification, below.

The sensor operand is to be computed at time TIME (milliseconds since time origin). If the environmental variables are available from tables as a function of millisecond of year, interpolation is performed using TIME + TFIXED as the argument.

One important consideration is protection against repeating rewinds. This is accomplished with the help of the variable J.

When the orbital data is input from an external I/O device, ORBIT is used as the input area.

In any future programming, the user should note that this module is called from module GADSFC. That is, it is called once per observation for each iteration. Inefficient coding may, therefore, result in greatly increased expenditure of computer time.

5.2 Specialization and Modification:

This module may be said to be specialized in that a definite environmental source is assumed, such as an orbit tape (or drum). Furthermore, in this module assume that the first two environmental variables of interest (sensor operands) are the solar line-of-sight and magnetic vectors and that their derivatives will not be required. The user may modify or add to these requirements. For an example, refer to GADSSO/EPED module designed for use with the EPE-D spacecraft.

In certain applications, the derivatives of the sensor operand with respect to the system parameters are needed. See Section VI. E. 2, Reference 1. The user programming these applications should insure that the derivatives are transmitted through a named common area in the same manner that the operands are transmitted; i. e., the same as ORBVEC is transmitted. As with ORBVEC, the customer programs are GADF01, 2. . . , GADG11, 12, 13. . . .

5.3 Order and Type of Interpolation:

In this module simple linear interpolation is used. If the user so desires, he may adopt more refined methods. See module GADSSO/EPED.

Module GADSSO/EPED. (Sensor operand for EPED)

1.0 Calling Sequence: CALL GADSSO.

2.0 Category: FORTRAN subroutine, GADS worker.

3.0 Purpose: To compute sensor operands for the case of the EPE-D satellite (S3-C).

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

The following variables are those which have not been discussed under module GADSSO. The reader is referred to that discussion for additional explanations.

CODR	Output. Cosine of RDEL.
EARTHR	Mean earth radius, kilometers.
IBALF	Address of BALF in tape input record.
IBDEL	Address of BDEL in tape input record.
IBTOT	Address of BTOT in tape input record.
IMFRP	Address of main frame period in tape input record.
IRALF	Address of RALF in tape input record.
IRDEL	Address of RDEL in tape input record.
IRDIS	Address of radial distance in tape input record.
JSCODE	Input. Type of environmental variable sought: 1 = solar, 2 = magnetic.
NBRECO	Number of orbit records read.

NUSEO	Flip-flop indicating which half of input area ORBIT is ready for use.
NWTRO	Number of words transferred in last read.
N28O	Number of words in one EPE-D orbit record.
RALF	Right ascension of orbit radius vector, radians.
RDEL	Declination of orbit radius vector, radians.
RDIS	Orbital radius distance.
RHO	Orbital radius vector, unit.
RDOTS	Vector dot product of RHO and unit solar line-of-sight vector.
SHADE	See explanation below.
SHADEC	See explanation below.
SIAR	Sine of RALF.
SIDR	Sine of RDEL.

5.0 Notes:

5.1 Calculation of ECLIPS:

To determine if the spacecraft is in the shade, the following calculations are made. First, the critical angle SHADEC is computed; this is illustrated in Figure 15a. Second, the actual shade angle is the arc-cosine of RDOTS (see Figure 15b). For an eclipse condition, SHADE is greater than SHADEC.

5.2 Format for Input Tapes to EPE-D Attitude:

5.3 General Description:

The tapes will be UNIVAC 1108 binary with fixed length records, i.e., 300 words per record. Each record will contain ID and orbit information and the channel 12, 14, 15, 0, 1 attitude data for 256 consecutive telemetry frames. The ID and orbit information will be contained in the first 44 words of each record in floating point format. The remaining 256 words contain the attitude data, all 5 channels for a given frame packed into a 36 bit words. The first frame recorded in each record will be of some PPO.

Each tape will constitute the data from twelve consecutive orbits. Each tape will contain one EOF following the last record.

The limitations on processable data will be as follows:

1. Must have sub-comm sync on PPO.
2. Must have at least 128 (out of possible 256) consecutive frames of data including no more than 10 padded frames of data. Missing frames which are padded should be checked using frame times at each end of the gap to insure the time difference is equal to NT within 10 milliseconds (or 10^{-7} day), where N is the number of missing frames and T frame period. Also positive PPO sync must be obtained on both sides of the missing data. The padded words and the filler words to completely fill the 256 words assigned to attitude data will be all zeros. All times will be in days and fractions of a day of the launch year, 1964. Thus 1200 UT on 31 December 1964 will be day 366.50 and 1200 UT on 1 January 1965 will be day 367.50.

5.4 <u>Record Contents</u>:

Word Number	Contents
1	Day of Year 1965
2	Fraction of Day
3	Average Frame Period in Fraction of Day

4-6	Coef. for Geographic Longitude (Radians)
7-9	Coef. for Geographic Latitude (Radians)
10-12	Coef. for Radial Distance (Km)
13-15	Coef. for Rt. Asc. of Subsat Point (Radians)
16-18	Coef. for McIlwain's L Parameter (Earth Radii)
19-21	Coef. for Magnetic Field Strength (Gamma)
22-24	Coef. for B local/B equator (ratio)
25-27	Coef. for Rt. Asc. of Magnetic Field (Radians)
28-30	Coef. for Dec. of Magnetic Field (Radians)
31	Station-Analog Tape Number (SSAAAA)
32	Date Recorded
33	Orbit Number
34	Status of File
35-44	Unassigned
45-300	A 256 word array where each word contains the channel 12, 14, and 15 HRP numbers and the channel 0 and 1 octal numbers for one frame of data. The structure of each word is as follows:

Bits	Information Contained
7-16	Channel 12 HRP number
17-26	Channel 14 HRP number
27-35	Channel 15 HRP number
1-3	Channel 0 octal number
4-6	Channel 1 octal number

All bad points, i.e., 999, 998, and 997 readings, will be signified by setting all bits assigned to the channel equal to zero.

5.5 Time and Orbit Parameters:

Word 2 is the start time of the first frame of data in the record minus one frame period. The start time of frame N (N = 1 to 256 is:

start time frame N = WORD2 + WORD3 x N in fraction of a day.

Similarly, the orbit parameters are given by three coefficients, such that: $PAR = C1 + C2 \times N + C3 \times N^2$.

For example, the L value for the start of the frame N is given by:

$L = WORD16 + WORD17 \times N + WORD13 \times N^2$ with L in earth radii.

5.6 <u>Reference:</u>

For further details, see Reference 18.

Module GADSSP. (Sensor parameters)

1.0 Calling Sequence: CALL GADSSP.

2.0 Category: FORTRAN subroutine, GADS executive, level 3.

3.0 Purpose: To prepare sensor-related parameters for efficient reference. Included are mounting constants, calibration (transfer function) constants, number of observations, sensor weighting, data starting address, etc.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

JS — Input. Sensor pointer index. The sensor being processed occupies the JS position in the current complex of sensors. Note definition of N0.

JSCODE — Output. Sensor operand code:

JSCODE = 1 - solar radiation,

JSCODE = 2 - geomagnetism,

JSCODE = 3 ,4 spares.

See also Run Deck, cards G13, G13-1, G13-2...

KSF — Output. Sensor ideal geometric operator code:

KSF = 1 - basic cartesian vector dot product,

KSF = 2 - basic cartesian vector dot product with shadowing,

KSF = 3, 4 spares.

See Run Deck, cards G13, G13-1, G13-2... Refer also to modules of the type GADK01.

KSO — Output. Calibration request flag used in module GADFC2.

KSW	Output. Special weighting request flag used in module GADSFC.
KU	System parameter index.
KU1	Output. Perturbed calculations indicator: KU1 = 1 normal, KU1 = 2 means perturbed calculations used in method of false position.
LENGTH(J)	Input. The length (size) of the array of observed data for sensor J.
LINKCP(J)	Output. Let the sensor being processed have its J calibration constant in the L position of the system parameter array. Then L = LINKCP(J). If L = 0, the said constant is not a member of the set of system parameters. See modules GOPER1 and KOPER1.
LINKF(J)	Input. The array of ideal prediction function codes for current complex. See modules GADSFC, GADSIN, and Run Deck, cards G13, G13-1, G13-2. . . . .
LINKNT(J)	Output. The array of NT values for current complex. For the J sensor in the given complex, NT = LINKNT(J) is the number of observed data points.
LINKOP(J)	Output. Same as LINKCP(J) but replace the words calibration constant with mounting constant or orientation constant.
LOCATE(J)	Input. Array of raw data starting addresses.
LOKATE	Output. Current starting address of raw data.
NCG	Input. Packed derivative (gradient) codes. See module GADSIN.

NOBDAT(J)	Output. Observable sensor-related codes:
	J=1 sensor operand generator type,
	J=2 ideal prediction function type f,
	J=3 transfer function (calibration) type $\mathscr{G}$,
	J=4 number of coefficients in $\mathscr{G}$,
	J=5 sensor operand type S,
	J=6 sensor mounting type (geometric operator) $\mathscr{H}$.
	Refer also to Section 3, Run Deck, card G13-7.
NT	Number of observed data points for a given sensor.
NU	Number of active system parameters.
NUMC(J)	Same as NUMCPJ.
NUMCPJ	Output. Packed form of codes described under variable LINKCP.
NUMOBS(J)	Input. Packed form of codes described under variable NOBDAT.
NUMOP(J)	Same as NUMOPJ.
NUMOPJ	Ouput. Packed form of codes described under LINKOP.
N0	Input. Pointer index used to identify sensor in permanent sensor library.
OBCOEF	Input. Library of observable (sensor) calibration coefficients. See Run Deck, card G13-8.
OBGAMM	Input. Library of sensor axis direction cosines. See Run Deck, card G13-7.
OBMOUN	Input. Library of observable (sensor) mounting constants. See Run Deck, card G13-7.
OBWGHT	Input. Library of sensor weighting. See Run Deck, card G13-9.

OCOEFF	Output. Current sensor's calibration constants in fast access area.
OGAMMA	Output. Current sensor's direction cosines in fast access area.
OMOUNT	Output. Current sensor's mounting constants in fast access area.
OTITLE	Input. Library of sensor identification (names).
SENSID	Output. Current sensor's name.
WEIGHT	Output. Current sensor's weight. See Run Deck, card G13-9.

5.0 Notes: Because references to the library of sensor data involve the use of indexes, unpacking, and some calculations, it is advantageous to move the pertinent data into special non-indexed locations in order to optimize the operations taking place inside tight loops. The purpose of this module is to answer this need in compact and centralized form. The variable N0 locates the required information in the permanent library of sensor data.

Module GADSSS. (Satellite simulation)

1.0 Calling Sequence: CALL GADSSS(COMMUT, WORK, NORBIT, NDIM, LMAX, NCALLS).

2.0 Category: FORTRAN subroutine, GADS accessory.

3.0 Purpose: To compute sensor outputs in a manner that simulates their behavior in real life. The simulation accounts for various types of motion, signal noise, signal bias, and characteristics of sampling by commutation.

4.0 Variables:

4.1 Explicit Inputs:

COMMUT	Commutator time delays in milliseconds for the various channels. See 5.1, Commutation.
WORK	Work area used in computation of PHI if NCASE = 3.
NORBIT	Number of complete orbital data sets.
NDIM	First dimension of WORK and the number of samples desired per sensor; i.e., the number of commutator turns. Also see Note 5.3.
LMAX	Number of sensors to be simulated.
NCALLS	Restart or re-initialization flag. When NCALLS = 0, all initial conditions will be reset. This flag should not be zero when returning to GADSSS for a second, third, ... pass of the same spacecraft; otherwise, the identities of the output functions will be sacrificed. For example, phase relations will be discontinuous. See Note 5.4.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

A	Input. First moment of inertia.
ADMH	Adams h, or step size, initially set to DELT1 but may be modified by module ADAMS, assuming DELT1 is a reasonable value for ADMH.
ALPHAI	Input. Right ascension angular momentum, degrees.
ALPHA	Right ascension, angular momentum, radians.
ALT	Input. Altitude of spacecraft in simulated orbit. At this writing, the orbit is circular. See module SORBIT.
ANGM(2)	Input. ANGM(1) = ALPHAI, ANGM(2) = DELTAI.
AOUTI	Hollerith P/O.
AOUTS	Hollerith P/O.
AUXF	Auxiliary functions and system state vector used by GADSAL.
B	Input. Second moment of inertia.
BDECL	Declination of B-field, radians.
BDECL1	Declination of B-field, degrees.
BDEL	Declination of B-field, radians.
BDELDG	Declination of B-field, degrees.
BL(3)	Cartesian components of B-field in local orbital coordinate system.
BRA	B-field right ascension, radians.
BRA1	B-field right ascension, degrees.
BTOT	B-field strength.

C	Input. Third moment of inertia.
COMM	Output. Common area for sensor output functions and sample times.
CONS1	Same as CONS, module GADA01.
COSTH	Cos (θ).
DECL	Declination of spacecraft, radians.
DECL1	Declination of spacecraft, degrees.
DELTA	Declination, angular momentum, radians.
DELTA1	Input. Declination, angular momentum, degrees.
DELT1	Input. Commutator cycle time and initial value of ADAMH.
DLAT	Spacecraft's latitude, degrees.
DLONG	Spacecraft's longitude, degrees.
DPR	Degrees per radian.
EPSIL1	See Note 5.3.
EPSIL2	See Note 5.3.
ERRMES	Hollerith P/O, error message.
FILLFL	Flat to signal fill-data.
FINC	Input. Orbital inclination to equatorial plane, degrees.
FINCL	Orbital inclination, radians.
FTEMP	Predicted ideal sensor output.
GDOT	Input. Greenwich rate, radians per millisecond.
GRWCH	Greenwich hour angle, radians.
GRWCHO	Input. Greenwich hour angle at TIME = 0.

ICOND	Input. Type of initial conditions:
	ICOND=1 - all balanced spacecraft,
	ICOND=2 - nonbalanced rigid spacecraft given the kinetic energy, magnitude of the angular momentum, and t_0 (see Reference 1, equations V. 8a, b, c, page 59),
	ICOND=3 - nonbalanced rigid spacecraft given angular velocity vector (see equations V. 3a, b, c, Reference 1),
	ICOND=4 - not used,
	ICOND=5 - not used.
IGNORE	Output. Sensor ignore flag.
IT	Pointer index for data sample.
JTIME	See module GADSCS.
JSCODE	Type of sensor operand code. See JSCODE in module GADSSP.
KBALF	Input. See module GADSSO.
KBDEL	Input. See module GADSSO.
KBTOT	Input. See module GADSSO.
KBX, Y, Z	Input. See module GADSSO.
KDAYOR	Input. See module GADSSO.
KECLPS	Input. See module GADSSO.
KHOURO	Input. See module GADSSO.
KORBNO	Input. See module GADSSO.
KOTAPE	Input. See module GADSSO.
KSALF	Input. See module GADSSO.

KSDEL	Input. See module GADSSO.
KSTOT	Input. See module GADSSO.
KSX, Y, Z	Input. See module GADSSO.
KWORDS	Input. See module GADSSO.
L	Input. See module GADSSO.
LENGTH(J)	Input. Length (size) of the array of raw data for sensor library.
LOCA	Input. Location of first raw data point in COMM.
LOCATE(J)	Input. Location of first raw data point for sensor J. It is part of the sensor library.
LOCT	Input. Locations of first raw data point sample time.
LSENS	Sensor pointer index, identical with N0.
MORBIT	Orbit item set counter or pointer.
MOTION	Type of motion requested: MOTION=1 - simple spin, MOTION=2 - Eulerian force-free precession, MOTION=3 - nonbalanced force-free motion, MOTION=4 - variation of parameters (not ready), MOTION=5 - numerical integration of equations of motion (not ready), MOTION>5 - not used. Note: MOTION is initially set equal to NCASE.
MSDORB	Input. See module GADSSO.
NCASE	Input. This is the initial value for MOTION.
NDEBUG	Input. Debugging aid. When non-zero, this flag will result in increased printout.

NGOTO	NGOTO=1 is normal. If NGOTO=2, a transformation of elliptic modules has occurred. See Note 4, module GADA01.
NOBDAT	Fast access area for observable sensor-related parameters.
NONCE1	Output. Referenced by module GADSAL.
NONCE2	Output. Referenced by module GADSAL.
NPARAM	Input. Array of special inputs. Refer to list of EQUIVALENCE statements.
N0	See LSENS.
N1	Location of current output data point in array COMM.
N4	Location of current data point sample time.
OGAMMA	Fast access area for sensor axis direction cosines.
OMEG	Input. Right ascension, orbital node, radians.
OMEGA	Same as OMEG, degrees.
ORBIT	Output. Array of orbital data.
ORBVEC	Orbit-related vectors (sensor operands).
PERIOD	Input. Not used at this writing.
PHDOTO	$\dot{\varphi}$ at time TSTART. See Figure 17 and Note 5.3.
PHI	φ, first Euler angle, radians.
PHIDOT	$\dot{\varphi}$, radians per millisecond.
PHIO	PHI at time TSTART, radians.
PROGID	Hollerith program identification.
PSI	ψ, third Euler angle, radians.
PSIDOT	$\dot{\psi}$, radians per millisecond.

PSIO	PSI at time TSTART, radians.
RA	Spacecraft's right ascension, radians.
RA1	RA in degrees.
RAD	Orbital radius vector.
RDOT	Orbital velocity vector.
REARTH	Mean earth radius, kilometers.
RPD	Radians per degree.
SIMMU	Simulated sensor noise mean value.
SIMSTA	Simulated sensor statistics: SIMSTA(1, J) - noise mean value for sensor J, SIMSTA(2, J) - noise standard deviation for sensor J.
SIMW	Simulated sensor noise standard deviation.
SPIN	Input. Spin rate: MOTION=1 - PSIDOT=SPIN, MOTION=2 - PHIDOT=SPIN, MOTION>2 - SPIN not used. See ZWMAGN.
STOT	Distance to sun. Not ready.
T	Sensor sample time since TFIXED. In preliminary PHI calculations, T is time since TPHIO.
TEMP	Collection of important variables. See EQUIVALENCE statements and initialization section.
TFIXED	Input. Double precision time origin, millisecond of year (or launch year, if desired).
THETA	θ, second Euler angle, radians.

TIME	Same as T.
TM	Input. Year of data. See Reference 9 and module FIELDG.
TORIGN	Input. Time-origin-related array of quantities. See coding of module GADSSO, EQUIVALENCE statements, and Section 2, Usage.
TPHIO	Time of PHIO, milliseconds.
TPREV	Time of channel 0 for current commutation cycle. This variable is called t_i in Note 1, below.
TRUEA	True anomaly, radians.
TRUEAR	Input. True anomaly rate, degrees per millisecond.
TRUER1	Same as TRUEAR initially but may be modified in module SORBIT.
TSTART	Input. Start time since TFIXED for first commutation cycle, milliseconds.
TWOPI	2π.
URN	Random number.
USC	Output. See module GADSLS.
VPARAM	Input. Array of important quantities. See EQUIVALENCE statements.
X	Spacecraft's orbital radius vector.
XW	Spacecraft's orbital plane radius vector.
YMILLS	Milliseconds per year, double precision.
ZALPHA	See equation 61.8, Reference 4.
ZAP	See equation 62.1, Reference 4.
ZARG	Argument of the elliptic functions.

ZBETA	See equation 61.4, Reference 4.
ZBQ	See equation 62.1, Reference 4.
ZCR	See equation 62.1, Reference 4.
ZG	Magnitude of angular momentum.
ZGAMMA	See equation 61.8, Reference 4.
ZGI	Input. Initial magnitude of angular momentum. See Note 4.
ZGSQ	Square of angular momentum magnitude.
ZINTGR	Temporary location for the integrand $\dot{\varphi}$/ZG.
ZK	The modulus of the elliptic functions.
ZP	x-component of angular velocity.
ZPO	ZP at time TSTART.
ZPOI	Input. Initial value of p. See Note 4.
ZPON	ZPOI normalized.
ZQ	y-component of angular velocity.
ZQO	ZQ at time TSTART.
ZQOI	Input. Initial value of q. See Note 4.
ZQON	ZQOI normalized.
ZR	z-component of angular velocity.
ZRO	ZR at time TSTART.
ZROI	Input. Initial value of r. See Note 4.
ZRON	ZROI normalized.
ZSIGMA	See equation 61.6, Reference 4.
ZT	Kinetic energy of rotation.

ZTI	Input. Initial value of kinetic energy. See Note 4.
ZTO	Input. Time when angular velocity lies in body x-z plane. See Figure 17 and Note 5.3.
ZTOS	Temporary storage for lower limit of integration in PHI calculations: ZTOS(1) = lower limit of integration adjusted for commutator delay, ZTOS(2) = not used.
ZWMAGN	Input. Magnitude of angular velocity at TSTART.

5.0 Notes:

5.1 Commutation:

In order to simulate the effect of a commutator, assume that the various sensors are sampled as follows: Let i and j refer to the ith sample time and jth sensor, respectively. Assume also that t_i is the time in milliseconds at which the commutator begins its ith revolution. Then the associated sample time for the jth sensor is t_i + COMMUT(j). This implies that the user must know the commutator time delay (or advance) incurred on each of the attitude sensing devices being simulated and should provide them through COMMUT. Refer also to the following paragraph.

5.2 How the Sensors are Chosen:

The module GADSSS will simulate all sensors registered in the sensor library by module GADSIN, beginning with the first and ending with the LMAX entries in the said library. To suppress the simulation of the J sensor, it is necessary to set LENGTH(J) = 0.

The sensors being simulated should have been successfully processed by GADSIN. Some inconsistencies in sensor definition can be detected during execution and can result in a sensor being ignored.

5.3 Preliminary Calculations of Precession Angle PHI for Non-balanced Spacecraft:

In the case of a force-free rigid non-balanced spacecraft (NCASE = MOTION = 3), the precession angle PHI is given by equation IV. 4f, Reference 1:

$$\varphi = \int_{t_0}^{t} \dot{\varphi} \quad dt' \tag{1}$$

where t_0 is some instant of time (specified by the user) when the angular velocity lies in the body x-z plane. (See Section V. E. 1, Reference 1.) Equation 1 is employed in this module and the integration is performed with module SIMP2.

Because SIMP2 works with arrays, the storage requirements may be excessive whenever $(t - t_0)$ is large. This can happen with the simulation of several passes of the same spacecraft. Hence, it is clear that some technique aimed at curbing core storage requirements is required.

The technique adopted in this module is to break up the integral (1) into two parts:

$$\varphi = \int_{t_0}^{t} = \int_{t_0}^{s} + \int_{s}^{t} . \tag{2}$$

where $t_0 \leq s \leq t$. Second, if s is the start time of the commutator for a given pass,

$$\varphi(s) = \int_{t_0}^{s} \tag{3}$$

is the precession angle at that time.

The preliminary calculation of equation (3) is, in turn, also broken up into several parts in the manner of equation (2), each part being just small enough to fit within the array WORK. That is, the integration is performed with data points taken in groups of size NDIM except the last group which will usually be smaller. The cumulative result of all integrations is φ (s) and is called ZPHIO. This precession angle, together with its associated time TPHIO, are saved until the next pass.

Inspection of the coding will reveal that a maximum of ten integrations is allowed between passes. That is, no more than ten groups of NDIM time steps of DELT1 milliseconds must elapse between passes. The user may increase this limit if it is desired. He should, however, consider the possible loss of accuracy in these calculations which depend on the accuracy of SIMP2.

As illustrated in Figure 17, if the forward integration is in increments of DELT1 the exact start time TSTART will not necessarily be reached. Hence, to obtain the precession angle at this time, proceed as follows.

Integration is halted the first time TSTART is exceeded. Then, to second order approximation (in the Taylor expansion sense), the excess in PHI is the product of the time derivative of PHI and the excess in time, viz

$$\Delta \varphi = \dot{\varphi} \Delta t,$$

or, in terms of coding,

EPSIL1 = PHDOTO*(T-TSTART).

5.4 Computing Initial Conditions:

Whenever the explicit input variable NCALLS is zero, initialization will occur. Initialization must be done at least on the first call to GADSSS. Successive calls will be considered successive passes of the same spacecraft if NCALLS $\neq$ 0. Otherwise initial conditions will be reset and phase relationships, etc., will be lost. These calculations depend on the type of motion desired, the latter determined by the variable NCASE.

(a) Simple Spin (NCASE = 1) - In this case, PSI, the third Euler angle, is used to denote spin and therefore PSIDOT is set equal to SPIN. SPIN, therefore, determines the spin rate in radians per millisecond. The user is to be certain that THETAI = 0. The orientation of the angular momentum is determined by means of ALPHAI and DELTAI.

(b) Eulerian (force-free) Balanced Precession - In this case, SPIN defines the precession rate PHIDOT in radians per millisecond. Then PSIDOT is determined using equation IV. 7b, Reference 1. Note that, if PHIDOT is defined in terms of PSIDOT, difficulty is encountered when cos (θ) = 0; i.e., when $\theta = 90^{\circ}$. As previous, the orientation of the angular momentum is determined by ALPHAI and DELTAI.

(c) Force-free Non-balanced Case (NCASE = 3) - As in the preceding cases, ALPHAI and DELTAI still specify the orientation of the angular momentum or auxiliary reference frame. (See Reference 1, Section V. E.) There are several ways to define the motion with respect to the reference frame, two examples are as follows:

Give the magnitudes of the angular momentum and kinetic energy of rotation, namely ZG and ZT. As explained in Reference 1, the motion is uniquely defined when the time t_0 is given. t_0 is defined above in Note 5. 3. To use this type of definition, the user should set ICOND=2.

Another method is to give the angular velocity vector $\Omega = \left| p, q, r \right|$. For convenience, the angular velocity components may be given by ZPOI, ZQOI, and ZROI. The magnitude is determined separately by ZWMAGN. To use this type of definition, the user should set ICOND=3.

Other methods of definition can be devised and the user may incorporate them using the above examples as a guide.

5.5 Jacobi Elliptic Functions:

For a discussion concerning the use of Jacobi's elliptic functions, refer to module GADA01. Much of the coding in these modules is identical.

5.6 Orbit-Related Data:

Assume that the attitude determination techniques being tested with the data generated by GADSSS make use of orbit-related parameters. Hence, an array of synthetic pertinent quantities is generated on FASTRAND for recall later. This simulates an orbit tape.

Module GADSST. (GADS statistics)

1.0 Calling Sequence: CALL GADSST(CORMAT, SVEC, EVEC, EIG, TC, CNORM, SP, XLEV, V, VNAMES, K1, K2, HEAD, SDEV, NV, NDEGF).

2.0 Category: FORTRAN subroutine, GADS accessory.

3.0 Purpose: Control module for calling all the subroutines used in computing statistics for GADS.

4.0 Variables:

4.1 Explicit Inputs:

CORMAT	Parameter correlation matrix.
CNORM	Vector whose components are used to de-normalize the correlation matrix. Equal to $1/\sqrt{d_{ii}}$, where d_{ii} is a diagonal element of the parameter covariance matrix.
V	Estimated values of the variables after the fitting process.
VNAMES	Mnemonic names of the variables.
HEAD	Heading to be printed on top of page for output.
SDEV	Standard deviation of the function; i.e., the square root of the residual variance.
NV	Number of active variables used in the fitting process.
NDEGF	Number of degrees of freedom. Equal to number of data points minus number of variables.

4.2 Explicit Outputs:

SVEC	Each column of this matrix represents an axis of the error ellipsoid in parameter-space, the center of the axis system being located at the estimated

parameter values. Each element of a particular column of SVEC represents a distance parallel to its associated parameter axis.

EVEC	Each column of EVEC is an eigenvector of the parameter correlation matrix CORMAT.
EIG	Associated eigenvalues of CORMAT.
TC	Ratio of each parameter to its standard deviation. Used to determine if a parameter is significantly different from zero at a given probability level.
SP	Standard deviation of the parameters.
XLEV	Probability level at which parameter is significantly different from zero.

4.3 Intermediate and Implicit I/O Variables:

CNTURS	Number of contours (probability levels) to be processed.
FA	Specific probability levels.
SC	Standard deviation of contours.

5.0 Notes: Reference 16.

Module GADSTV. (Displays)

1.0 Calling Sequence: CALL GADSTV (JJJJ).

2.0 Category: FORTRAN subroutine, GADS executive, level 2.

3.0 Purpose: To generate graphic displays on the printer or the SC 4020.

4.0 Variables:

4.1 Explicit Inputs:

JJJJ	Control bits for displaying text of numerical results: JJJJ=8 - display text, JJJJ≠8 - do not display text. (The text includes results of differential correction and the time.)

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

ABSCIS	Abscissa, temporary location.
ABSC1	Abscissa, current A value (observed or raw data value).
ABSC2	Abscissa, next A value.
ACHAR	Library of display character codes for A values for each sensor. See Run Deck, card G13-6.
AVALUE	A value (observed raw data).
COMM	I/O and work space. See module GADSCS.
CTEMP	Calibrated predicted output, temporary location.
DX1, DX2	Grid line spacers, vertical grids.
DY1, DY2	Grid line spacers, horizontal grids.
ECHAR	Standard error character code.
FCHAR	Predicted function, ideal, character code.

FDISP	Library of plotting controls. See Run Deck, card G15.
FILLFL	Fill flag. Fills are identified by -0 and are ignored.
FTEMP	Predicted function, ideal, temporary location.
IA, IB	See KGRAPH.
IBIT	2^{18} or 262144.
ICALIB	Calibration request: 1 = no request 2 = request calibration of current sensor.
IDENS	See IDENS1, 2.
IDENS1	Maximum number of dense predicted points. See Run Deck, card G15.
IDENS2	Maximum number of dense predicted points. See Run Deck, card G15.
IGNORE	Ignore current data flag.
IGRPH1	Display control parameter. Controls labelling. See Run Deck, card G15.
IGRPH2	Display control parameter. Controls labelling. See Run Deck, card G15.
IT	Data pointer index.
JA	See KGRAPH.
JGRPH1	See IGRPH1.
JGRPH2	See IGRPH1.
JS	
JTIME	See variable JTIME, module GADSCS.
KA	See KGRAPH.

KGRAPH — Input. Display control bits. Beginning with the low-order half of the word, the purpose of each bit is as follows:

Power of 2	Variable Name	Purpose
0	(none)	Display 1st participating sensor
1	(none)	Display 2nd participating sensor
2	(none)	Display 3rd participating sensor
.		
.		
.		
15	(none)	Display 16th participating sensor

The following bits qualify the preceding requests. Beginning with the high-order half of the word, the purpose of each bit is as follows:

Power of 2	Variable Name	Purpose
0	1A	Display raw data
1	JA	Display ideal predicted values
2	KA	Display "dense" ideal predicted values
3	LA	Display "$\pm$ 1 sigma" envelope
4	MA	Display modified predicted values (calibrated values), that is, display predicted telemetry counts.
5	NA	Display residuals

The following bits, in turn, further qualify the preceding requests:

6	IB	Superimpose all sensor curves, that is, let all sensor share the same display space

	7	Ignored at this writing
	8	Ignored at this writing
	9	Display only after final results of differential correction
	10	Display each time convergence in differential correction is achieved
	11	Display each time a differential correction is applied

Note: Display requests 10 and 11 may result in a large amount of output and should be used with care. These requests are intended for special trouble-shooting purposes only.

KSO	Calibration request indicator: KSO=0 - no request, KSO=1 - calibration requested. See Section 3, Run Deck, card G14d.
LA	See KGRAPH.
LAUXG	Relative location of auxiliary functions.
LDEQF	Relative location of (differential equation) integrated functions. See module ADAMS.
LDISP	Library of plotting controls. See Run Deck, card G15.
LGRPH1	Frame advance control.
LGRPH2	Frame advance control.
LOCATE	Input. See paragraph 2.6.
LOCEND	Input. End of available work area.
LOCFLG	Input. Relative location of FLG array used in module GADSFC.

LOCF1	Input. Relative location of first predicted function.
LOCTOP	Input. Highest location of work area assigned by module GADSCS.
MGRPH1	Horizontal grid emphasis control.
MGRPH2	Horizontal grid emphasis control.
NAUX	Input. Number of auxiliary functions.
NCALLS	I/O. Call counter.
NCUT	Input. Number of false position derivative.
NDEQ	Input. Number of differential equations being integrated under module GADSDC.
NS	Input. Number of participating sensors.
NT	Input. Number of data samples for a given sensor.
NTNS	Input. The product NT*NS.
NXGRPH	Vertical label control.
NYGRPH	Horizontal label control.
N0	Pointer index used to identify sensor in permanent sensor library.
N1	Location of predicted function.
N2	Location of raw data.
N3	Location of residual.
N4	Location of sample time.
N6	Location of flag word used during outlier detection.
PARAMS	Input. Hollerith parameter names.
POWERS	Input. Powers of 2, integer.

PROGID	Hollerith program identification.
SDEV	Input. Standard deviation of fit. See module GADSLS and GADSST.
SENMAX	Input. Permanent library of maximum vertical deflection. For example, SENMAX(N0,I); I=1,2 yields the maximum for ideal prediction function and modified (calibrated) prediction function, respectively.
SENMIN	Same as above except for minimum vertical deflections. See Run Deck, card G13-6.
TFIXED	Input. Double precision time origin, in milliseconds of year.
TIME	Sample time since time origin, in milliseconds.
TLABEL	Input. Hollerith labels for time printout. See module GADSDA.
TORIGN	Input. Time origin array. Refer to Section 2, Usage.
USERID	Input. Hollerith user identification. See Run Deck, card G1.
U	Input. System parameters. U corresponds to UTEMP in module GADSLS.
V	Input. Hollerith blanks or asterisks, depending on whether the corresponding parameter in U was refined.
XL	Horizontal left limit.
XR	Horizontal right limit.
XY	Horizontal left limit (to be loaded by module GRID1V only).
XZ	Horizontal right limit (to be loaded by module GRID1V only). Note: The reason for preventing GADSTV

from disturbing locations XY and XZ is apparent upon consideration of module PAPER. This module requires the left and right display limits of the previous frame.

YB	Lower vertical display limit.
YT	Upper vertical display limit.

5.0 Explanation:

5.1 Important Considerations:

The main considerations in this module are to analyze JJJJ immediately upon entry (if necessary), to analyze KGRAPH, and to generate the executive steps needed to obtain the desired displays.

JJJJ is used to infer high-level branching decisions. If, in future use, this variable is to be used to signal certain types of displays not yet programmed, use JJJJ code values greater than 32. KGRAPH provides additional detail. The lower half of this word determines which sensors are to be displayed. The high-order half contains details concerning the type of data to be displayed. See Section 4.3, page A-165. In addition, it also indicates if curves are to be superimposed. The user may, likewise, create a variable such as KGRAPH in a named common area, if he wishes to add displays of his own.

5.2 Internal Subroutines:

In order to make efficient use of the code, the present module contains several internal FORTRAN subroutines. Subroutine GRAPH1 displays all sensor curves taking one sensor at a time. Subroutine GRAPH2 displays all curves taking one frame at a time, i.e., superimposing the curves of the various sensors.

Subroutine GRAPH3 calculates the various addresses needed to locate the pertinent data in work areas. GRAPH4 loads the pertinent character codes from the library. GRAPH5 is used to generate intermediate predicted sensor output functions, i.e., between actual sample times. This is done when dense curves are to facilitate the interpretation of the output. Finally, GRAPH6 is used to display ± 1 sigma error envelope.

5.3. Film vs. Paper Output:

When the user deletes the GADS modules GRID1V and POINTV from the program complex prior to execution, the GSFC system modules will be invoked. The result is an SC 4020 tape from which film or hardcopy can be made. Otherwise, the plotting will be done on the high-speed printer.

Module GADW01. (Weighting function, type 1)

1.0 Calling Sequence: CALL GADW01.

2.0 Category: FORTRAN subroutine, GADS worker.

3.0 Purpose: To calculate the weighting function to be applied to the squared error.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

TIME	Input. Millisecond since time origin (or since base time) for which weighting function is desired.
WEIGHT	Output. The desired weighting.

5.0 Notes: Normally the user specifies a constant weighting function by means of the sensor definition cards of the type G13-8. This results in a constant WEIGHT value for the sensor in question. See Run Deck, card G13-8.

When more complicated weighting is needed, the options explained in cards G14a-2, 3 override the previous logic. As explained in the card description, the user may specify the number, for example, N, of a module of the type GADW01. The value of N may be:

N=0 - use regular weighting as in card G13-8,

N=1 - use module GADW01 to compute WEIGHT,

N=2 - use module GADW02 to compute WEIGHT,

. . . .

N=8 - use module GADW08 to compute WEIGHT.

As explained in Section III. A, pp. 13, 14, Reference 1, the desired weighting functions may be complicated functions of time. They may also be functions of the environmental variables.

Module ADAMS. (Adams-Bashford integration)

1.0 Calling Sequence: See below.

2.0 Category: SLEUTH II subroutine, GADS support.

3.0 Purpose: To integrate a set of N first-order simultaneous differential equations using the Adams-Bashford variable step method.

4.0 Variables:

4.1 Explicit Inputs:

N	Number of first-order differential equations to be integrated.
E	Desired accuracy. For example: E = .001 - 3-place accuracy, E = .0001 - 4-place accuracy, ...
X	Independent variable or abscissa.
XP	Value of X where P is to be evaluated.
H	Variable integration step size. H should be set initially to a reasonable value.
F(N)	Array of N, initial conditions $y_1(0), y_2(0), \ldots y_N(0)$. See examples below.
D(N)	Array of N derivatives $\frac{dy_1}{dx}, \frac{dy_2}{dx}, \ldots \frac{dy_N}{dx}$. See samples below.

4.2 Output Variables:

P(N)	Array of N integrated functions $y_i(XP)$. Note that these quantities are obtained by interpolation from the F's.
F(N)	Array of N integrated functions $y_i(X)$.

4.3 Intermediate Variables:

Z(8N)	Array of work space of size 8N.
H	Variable step size. See 4.1, Explicit Inputs.

5.0 Notes:

5.1 Usage: The first step in using this program package is to initialize. This is accomplished with the following statement: CALL ADMSET (N, F, D, P, E, Z, X, H).

To integrate forward a single step, use the following sequence of operations:

a) CALL ADMINT

b) compute the derivatives D

c) CALL ADMCOR

Step a) will predict values of y_i; i = 1, 2, ... N at X+H and the last step will correct. Upon return from ADMCOR, the new values y_i(X+H); i = 1, 2, ... N, are received.

Next, to obtain the values y_i at a point XP, use the statement

CALL ADMPAR(XP).

Upon return from this call, the array P will contain the desired values y_i(XP). This will not upset F, D, X, or H.

In order to restart the integration at a discontinuity,

CALL ADMRES.

The difference table will be cleared and can be changed.

5.2 Example:

Assume the system to be integrated is:

$$y_1 = e^x$$

$$y_2 = - \sin x$$

$$y_3 = \cos x.$$

Let the initial conditions be:

$$y_1 = F(1) = 0,$$

$$y_2 = F(2) = 0,$$

$$y_3 = F(3) = 0,$$

and integrate from $x = 0$ to $x = 10$, printing the results at $x = 0, 2, 3, \ldots 10$. The following is a suitable program:

```
      DIMENSION Z(24), F(3), D(3), P(3), E(3)
C SET INITIAL CONDITIONS
      DO 2 I= 1,3
    2 F(I) = 0
      X = 0
      DP = 0
      DT = .2
      DF = 10
      H = .001
      DO 3 I = 1,3
    3 E(I) = .0001
C INITIALIZE
      CALL ADMSET (3,F,D,P,E,Z,X,II)
C INTEGRATE FORWARD
      1 CALL ADMINT
```

```
C CALCULATE DERIVATIVES

      D(1) = EXP(X)
      D(2) = -SIN(X)
      D(3) = COS(X)

C CORRECT

      CALL ADMCOR

C TEST FOR PRINT  POINT

      3  IF (X-DP) 1,1

C INTERPOLATE

      CALL ADMPAR(DP)

      PRINT 10,  DP,  P

   10  FORMAT (1X,  4E18.8)

      DP = DP + DT

      IF (DP-DF) 3,3

      STOP

      END
```

5.3 References: For further discussion about the Adams-Bashford partial step method, see References 6 and 7.

Modules CN, DN, and SN. (Elliptic functions)

1.0 Calling Sequence: See Usage.

2.0 Category: FORTRAN function, GADS accessory.

3.0 Purpose: To compute the Elliptic Functions of Jacobi. If the elliptic integral of the first kind, $F(k, \varphi) = u$, the inverse function is called the amplitude function $\varphi = \mathrm{am}(u, k)$. The Jacobian elliptic functions are defined by:

a) $\mathrm{sn}(u, k) = \sin \mathrm{am}(u, k)$

b) $\mathrm{cn}(u, k) = \cos \mathrm{am}(u, k)$

c) $\mathrm{dn}(u, k) = (1-k^2 \, \mathrm{sn}^2 \, (u, k))^{1/2}$

4.0 Variables:

4.1 Input Variables:

U	The argument or parameter.
XK	The modulus, k.

4.2 Output Variables: The functions CN, DN, and SN each return a single variable according to the rules of FORTRAN functions.

5.0 Notes:

5.1 Method: The formulae are based upon Gauss' theory of the arithmetico-geometrical means and Legendre's method of computation. If a_o and b_o are two positive numbers then the sequences

$a_{n+1} = \frac{a_n + b_n}{2}$ and $b_{n+1} = (a_n b_n)^{1/2}$ converge to the same limit,

the arithmetico-geometrical mean of a_o and b_o.

For $k < 0.9539$ with $a_o = 1$ and $b_o = \sqrt{1-k^2}$ and

$$a_{n+1} \cot \varphi_{n+1} = 1/2 \left(a_n \cot \varphi_n - \frac{a_n b_n}{a_n \cot \varphi_n}\right) \text{ and with } \frac{\varphi_4}{a_4} \cong u = F(k, \varphi)$$

then $\sin\varphi_o = sn(u,k)$ is found by

$$\sin\varphi_i = \frac{2a_i \sin\varphi_{i+1}}{(a_i+b_i) + (a_i - b_i)\sin^2\varphi_{i+1}} \quad .$$

When $\varphi_4 \leqq \frac{\pi}{4}$ then $\varphi_o \leq \frac{\pi}{4}$, we use $cn(u,k) \cong (1-\sin^2\varphi_o)^{\frac{1}{2}}$.

When $\varphi_4 > \frac{\pi}{4}$

$$cn(u,k) \cong \frac{\sqrt{1-k^2}\ \sin\varphi_o^*}{((1-k^2)\sin^2\varphi_o^*)^{\frac{1}{2}}}$$

and $\sin\varphi_o^*$ is obtained from $\varphi_4^* = \frac{\pi}{2} - a_4 u$.

For dn, $dn(u,k)$, $\cong (1-k^2\sin^2\varphi_o)^{\frac{1}{2}}$ for $\varphi_4 \leq \frac{\pi}{4}$

$$dn(u,k) \cong \frac{\sqrt{1-k^2}}{(1-k^2\sin^2\varphi_o^*)} \quad \text{for } \varphi_4 > \frac{\pi}{4}$$

When $k \geq 0.9539$, $\tan\varphi_3 = \sinh(a_3 u)$, then $\tan\varphi_o$ can be calculated with

$$\tan\varphi_i = \frac{2a_i \tan\varphi_{i+1}}{(a_i+b_i) - (a_i-b_i)\tan^2\varphi_{i+1}}$$

Then

$$sn(u,k) = \sin\varphi_o \cong \frac{\tan\varphi_o}{(1+\tan^2\varphi_o)^{\frac{1}{2}}}$$

$$cn(u,k) \quad \frac{1}{(1+\tan^2\varphi)^{\frac{1}{2}}}$$

$$dn(u,k) \cong \left(\frac{1+(1-k^2)\tan^2\varphi_o}{1+\tan^2\varphi_o}\right)^{\frac{1}{2}} .$$

5.2 Usage:

SN(U, XK) in a floating point arithmetic expression.

CN(U, XK) in a floating point arithmetic expression.

DN(U, XK) in a floating point arithmetic expression.

5.3 Accuracy: Relative truncation error of the order 10^{-12}.

5.4 Restrictions:

$0 \leq XK \leq 1$ and $0 \leq u \leq F(k, \frac{\pi}{2})$

If $XK < 0$, the routine uses $|XK|$

If $XK > 1$, the routine sets the function to -1.

Module EOFTV. (End file, display)

1.0 Calling Sequence: CALL EOFTV.

2.0 Category: FORTRAN subroutine, GADS support.

3.0 Purpose: To replace GSFC's module of the same name when display is desired on the high-speed printer.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables: None.

5.0 Explanation:

At the conclusion of a run, the last frame (page) of display remains to be output. Hence, this module is designed to provide the last call to PAPER whose function is to generate a frame (page) of output.

Module EPEDIN. (EPE-D input preprocessor)

1.0 Calling Sequence: CALL EPEDIN(TDATA, ITDATA, F, T, TSTART, NFNR, $).

2.0 Category: FORTRAN subroutine.

3.0 Purpose: To preprocess input data obtained from the program described in Reference 18.

4.0 Variables:

4.1 Explicit Inputs:

TDATA	Tape input data to be processed.
ITDATA	Same as TDATA.
NFNR	Total number of frames to be processed and the dimension of F and T. Note: This number, computed in GADS/EPED, is the product of the number of frames per record and the number of records brought in off tape.
	FORTRAN abnormal return address in case data is not up to standards. See Note 1.

4.2 Explicit Outputs:

F(NFNR,6)	Sensor output functions for use in attitude determination. These may or may not be calibrated to engineering units. See variable ICALB below.
T(NFNR,6)	Observation times corresponding to the values F. Time is measured in milliseconds and referenced to TSTART.
TSTART	Double precision millisecond of launch year 1965. This module will compute TSTART only when variable NONCE = 0. See Note 2.

4.3 Intermediate and Implicit I/O Variables:

ADAY	Output. Day of launch year.
ADAY1	Output. Previously computed ADAY.
AHOUR	Output. Hour of day.
AMILLS	Output. Millisecond of minute.
AMINUT	Output. Minute of hour.
BACFOR	Integer pointer indexes used in computing theta angles. See Note 3.
DAYNG	Hollerith message.
DELT	Mean frame period in milliseconds.
DELTCH	Mean channel period in milliseconds.
DMILLS	Milliseconds in a day: 86400000.
EPCOEF	EPE-D calibration coefficients. See Note 4.
I, II, III	Temporary indexes. See Note 5.
LARG	Argument for table look-up in theta angle calibration. See Note 3.
ICALB	Input. Calibration request: ICALIB(I) = 0 - no calibration, ICALIB(I) = 0 - calibrate, where I = 1 - X magnetometer, I = 2 - Z magnetometer, I = 3 - solar patch.
IFORBA	See Note 3.
INSYNC	See Note 5.
IOCTAL	See Note 5.

IREQST	Input. Requested locations for various sensor outputs:
	IREQST(1) - location for solar patch,
	IREQST(2) - location for Z magnetometer,
	IREQST(3) - location for X magnetometer,
	IREQST(4) - ignored,
	IREQST(5) - location for theta angles.
	Note: The term location is the second index of F and T.
IT	Pointer index for F and T data.
IUNITS	See Note 3.
J	Pointer index for sensor output data.
K, KK, KKK	See Note 5.
KOUNT	Counter for zero data. See Note 1.
L	Pointer index for channel to be processed:
	L = 1 - process channel 15 (solar patch)
	L = 2 - process channel 14 (Z magnetometer)
	L = 3 - process channel 12 (X magnetometer)
	L = 4 - process channel 16 (or 0) (solar aspect)
	L = 5 - process channel 1 (solar aspect).
LOOKN3	See Note 3.
MODLUS	See Note 6.
NCALLS	Call counter.
NCOEF	Input. Number of coefficients in calibration formulas:
	NCOEF(1) - number of coefficients, X magnetometer,
	NCOEF(2) - number of coefficients, Z magnetometer,
	NCOEF(3) - number of coefficients, solar patch.

NONCE	Input. When NONCE = 0, EPEDIN-4 will reset TSTART and signal the beginning of a new pass. Note that this variable is called NONCX in module GADS/EPED.
N1, N2, N3	Temporary working cells used in computing theta angles. See Note 3.
THCALB	Calibration table for theta angles. See Note 3.
TIJ	Data sample time relative to TSTART, milliseconds. Note that channels 12, 14, and 15 are adjusted for commutator delays.
TORIGN	See module GADSLS.
TPREV1	Double precision (previous) previous frame time relative to TSTART, milliseconds.
TPREV2	Double precision previous frame time relative to TSTART.
ZFRMP	Hollerith message indicating zero frame period. See Note 1.

5.0 Notes:

5.1 Data Standards: The "EPE-D Attitude and Orbit Tape Extract Program," Reference 18, may occasionally produce zeros. This signals poor data quality. In such cases the data should be discarded. When TDATA(3) = 0 (the mean frame period), the entire record should be discarded. When an individual sample is zero, only that sample should be discarded.

5.2 Resetting the Time Origin: Normally EPEDIN will be invoked several times during the one data pass. Hence, the time origin TSTART should not be disturbed except to start a new pass. The calling program GADS/EPED causes TSTART to be reset by NONCE = 0.

5.3 <u>Optical Aspect Calculations:</u> Optical aspect data are processed according to the instructions in Appendix 1, Reference 20. These instructions are as follows:

"Channel 0 and channel 1 contain the EPE-D optical aspect data. To reduce solar aspect data, perform the following steps:

1. Examine channel 1 for a value other than zero. Call this value N3.
2. Call the value of channel 0 immediately preceding N3, by the name N2. Note: The telemetry transmission order of channels is 1, 2, 3, ... 15, 0.
3. Call the value of channel 0 preceding N2 by the name N1.

SEE-SUN TIME EXTRACTION

The value of N1 is a time vernier. It identifies the particular one-eighth of the frame immediately preceding N1 during which the sun appeared in the plane of the aspect sensor. For example:

If N1 = 100, the sun appeared during channels 1 and 2 prior to N1.

If N1 = 200, the sun appeared during channels 3 and 4 prior to N1, and so on.

If N1 = 000, the sun appeared during channels 15 and 0 (of which N1 is channel 0); i.e., the sun appeared during readout of N1.

SUN ANGLE EXTRACTION

The value of N2 and N3 provide a measure of the angle (θ) between the satellite spin axis and the incident sunlight at the time given by N1."

The calibration talbe for the theta angles is shown in the same reference. It is included in this program and stored in the array THCALB in the form of constants. Unfortunately, the method of determining theta from N2 and N3 is not straight-forward. In this module, a pointer index LARG to retrieve the required angle from the table is computed. LARG can be assumed to be a two digit number in the octal system. The low order digit is called IUNITS; the high order, IOCTAL.

As illustrated in Table 1-1, Appendix 1, Reference 20, IUNITS is determined by N2, IOCTAL by N3. In addition, there is a correspondence between N3 and the digits 0 through 7, as follows:

N3 = 0	corresponds to	0
N3 = 0	corresponds to	0
N3 = 1	corresponds to	1
N3 = 3	corresponds to	2
N3 = 2	corresponds to	3
N3 = 6	corresponds to	4
N3 = 7	corresponds to	5
N3 = 5	corresponds to	6
N3 = 4	corresponds to	7.

N2 works in the same way when N3 = 0, 3, 6, and 5. In the other cases, however, N2 works in reverse order. To compute IARG, it is necessary to transform N3 according to the foregoing table and transform N2 according to the same table but taking into account the fact that it sometimes works backwards, as stated above. The variable IFORBA determines whether to go backwards or forward by means of the digits 2 and 1, respectively. These indicators are stored in array BACFOR.

5.4 Calibration Coefficients: Calibration coefficients are provided by the user through variables EPCOEF and NCOEF which are read from cards in program GADS/EPED. The formula is always assumed to be of the form:

$$y = \sum_{i=2} C_i \, (x + C_1)^{i-1}$$

where x = telemetry count

y = engineering value

C_i = $i^{\underline{th}}$ calibration constant,

n = number of constants.

In order to obtain n and the value of C_i for the $j^{\underline{th}}$ sensor, then,

n = NCOEF(J)

C_i= EPCOEF(I, J).

The sensors are numbered so that:

J = 1 - corresponds to X magnetometer,

J = 2 - corresponds to Z magnetometer,

J = 3 - corresponds to solar patch.

The optical sensor calibration is automatically performed separately, see above.

5.5 Programming:

Because this module was originally written to extract the data from the EPE-D Master Digital Tapes (Reference 18), several of the original varible names remain vestigial. These include II, III, IIII, KK, KKK, KKKK, and INSYNC.

5.6 Unpacking the Input Word:

Since the input data arrives packed into one 1108 word, it is necessary to strip the bits according to the rules given on page i, reference 18. This is accomplished by repeated remaindering. The divisors are stored in array MODLUS.

Module FDIST. (Snedecor F distribution)

1.0 Calling Sequence: CALL FDIST(P, F1, F).

2.0 Category: FORTRAN subroutine, GADS accessory.

3.0 Purpose: To compute the F statistic for a given probability level as a function of the two specified number of degrees of freedom.

4.0 Variables:

4.1 Explicit Inputs:

P	The probability level for which the F statistic is to be computed.
F1	The number of degrees of freedom for the first variable.
F2	The number of degrees of freedom for the second variable.

4.2 Explicit Outputs:

F	The F statistic as a function of the specified variables.

5.0 Notes: The subroutine computes the value of the Snedecor F_γ for F1, F2 degrees of freedom, where γ is the probability level.

Module FDISTE. (F distribution for F1 even integer)

1.0 Calling Sequence: CALL FDISTE(P, F1, F).

2.0 Category: FORTRAN subroutine, GADS accessory.

3.0 Purpose: To compute the F statistic for a given probability level when the number of degrees of freedom of the first variable F1 is an even integer.

4.0 Variables:

4.1 Explicit Inputs:

P	The probability level for which the F statistic is to be computed.
F1	The number of degrees of freedom for the first variable.
F2	The number of degrees of freedom for the second variable.

4.2 Explicit Outputs:

F	The F statistic as a function of the specified variables.

Module FIELDG. (FIELDG simulator)

1.0 Calling Sequence: CALL FIELDG (DLAT, DLON, ALT, TM, N, J, B1, B2, B3, B).

2.0 Category: FORTRAN subroutine, GADS accessory.

3.0 Purpose: To simulate the calculations performed by FIELDG program documented in Reference 9.

4.0 Variables:

4.1 Explicit Inputs:

DLAT	Degrees latitude (not used)
DLON	Degrees longitude (not used)
ALT	Altitude, kilometers (not used)
TM	Time in years (not used)
N	Maximum degree of expansion plus one (not used)
J	Control variable (not used)

4.2 Explicit Outputs:

B1	North component of field
B2	East component of field
B3	Vertical component of field (positive down)
B	Total field (magnitude) Units of F are same as coefficients

4.3 Intermediate Variables:

BX	North component examples
BY	East component examples
BZ	Vertical component examples
B	Total field
L	Call counter
M	Example pointer

5.0 Note: This module is a dummy or stand-in for FIELDG documented in Reference 9. For definitive simulation the original FIELDG should replace this module. For quick checkout purpose, however, this fast and compact version is adequate. Also note that the authors of Reference 9 have developed improved versions since 1964.

Module GENORM. (Normally distributed random number generation)

1.0 Calling Sequence: CALL GENORM (AV, W, VAR).

2.0 Category: FORTRAN subroutine, GADS accessory.

3.0 Purpose: To generate random numbers with approximately normal distribution.

4.0 Variables:

4.1 Explicit Inputs:

AV	Desired mean (average)
W	Desired standard deviation

4.2 Explicit Outputs:

VAR	Random variate with mean AV and standard deviation W.

5.0 Notes:

5.1 Usage:

The output variable is VAR and the user must supply AV and W as described above. Only one VAR is generated per call. GENORM can be used for generating distributions with different W and AV without regard to order.

5.2 Mathematical Note:

Using the subroutine RANDOM, first obtain individual samples X_i from a set of random numbers uniformly distributed in the range 0-1. The set has a mean one-half and standard deviation of one-twelth since the first moment (mean) is:

$$\mu = \int_{-\infty}^{\infty} P(x)\ XDX = \int_{0}^{1} XDX = 1/2$$

and the second moment about μ (variance) is:

$$\sigma^2 = \int_{-\infty}^{\infty} (X-\mu)^2 P(X)\, DX = \int_0^1 (X - \mu)^3 \, DX$$

$$= \left(\frac{X^3}{3} - 2\frac{X^2 \mu}{2} + X\mu^2 \right) \int_0^1 = 1/12$$

Then a new random sample is formed from six individual samples.

First consider

$$\varphi = \frac{I}{6} \sum_{i=1}^{6} (X_i - \mu)$$

which is a sample from an approximately normal distribution and has, by the Central Limit Theorem, a standard deviation,

$$\sigma_\varphi = \frac{\sigma}{\sqrt{6}}$$

Observe that a new random variable with unity variance can easily be generated:

$$\Theta = \frac{1}{6} \sum_{i=1}^{6} \frac{\sqrt{6}}{\sigma} (X_i - \mu),$$

where the distribution has the property $\sigma_\varphi = 1$.

A sample can now be obtained from a set having a desired standard deviation, W, by multiplying a generated value of Θ by W.

A new mean, AV, can be introduced into the distribution by adding AV to each individual sample. Therefore, the final formula for the program variable VAR is

$$VAR = W \sum_{i=1}^{6} [X_i - 1/2] \sqrt{2} + AV.$$

Module GRID1V. (G. S. F. C. GRID1V simulation)

1.0 Calling Sequence: CALL GRID1V(LPL, XL1, SR1, YB1, YT1, DX1, DY1, NPL, MPL, IPL, JPL, NXPL, NUPL).

2.0 Category: FORTRAN subroutine, GADS support.

3.0 Purpose: To replace GSFC's module GRID1V when displays are preferred on the high-speed printer.

4.0 Variables:

4.1 Explicit Inputs:

LPL	Ignored.
XL1, XR1	Horizontal left and right limits of the grid.
DX1	Vertical grid line spacer.
DY1	Horizontal grid line spacer.
NPL, MPL	Ignored.
IPL, JPL	Ignored.
NXPL, NYPL	Ignored.

4.2 Explicit Outputs: None

4.3 Intermediate and Implicit I/O Variables:

BLANKS	Hollerith blanks.
DELX	Output. Horizontal grid size or span.
DELY	Output. Vertical grid size or span.
DX	Output. Vertical grid line spacer.
DY	Output. Horizontal grid line spacer.

EJNOT	Hollerith printer control, namely 6H+. See Module GADSDV.
NCALLS	Call counter.
PRARR	Output. Print array.
XFACTR	Output. Horizontal conversion to raster counts factor.
YFACTR	Output. Vertical conversion to raster counts factor.
XL	Output. Left limit of grid.
XR	Output. Right limit of grid.
YB	Output. Bottom limit of grid.
YT	Output. Top limit of grid.

Module HDIAG. (Diagonal of Hermitian matrix)

1.0 Calling Sequence: CALL HDIAG(NV, IVECF, NR, CORMAT, SVEC, EVEC, IQ).

2.0 Category: FORTRAN subroutine, GADS accessory.

3.0 Purpose: To compute the eigenvalues and eigenvectors of a real symmetric matrix.

4.0 Variables:

4.1 Explicit Inputs:

NV	Number of variables in the fitting process; i.e., the order of the matrix CORMAT.
IVECF	IVECF=0 is a flag to compute both eigenvalues and eigenvectors. IVECF=1 is a flag to compute only eigenvalues.
CORMAT	Parameter correlation matrix for which the eigenvalues and eigenvectors are to be computed.

4.2 Explicit Outputs:

NR	The number of rotations.
EVEC	Matrix containing the eigenvectors.
SVEC	Temporary storage vector.
IQ	Temporary storage vector.

5.0 Notes: This subroutine computes the eigenvalues of eigenvectors of a real symmetric matrix using the Jacobi method. CORMAT is diagonalized, the diagonal elements being the eigenvalues. Only the elements of CORMAT that are to the right of the main diagonal are operated upon, and only this triangular section need be passed to the subroutine.

Module INNERP. (Vector inner product)

1.0 Calling Sequence: CALL INNERP (A, B, C).

2.0 Category: FORTRAN subroutine, GADS accessory.

3.0 Purpose: To calculate the inner product, or dot product, of two cartesian vectors.

4.0 Variables:

4.1 Explicit Inputs:

A(3)	First cartesian vector.
B(3)	Second cartesian vector.

4.2 Explicit Outputs:

C	Inner product A * B.

Module INTERP. (Linear interpolation)

1.0 Calling Sequence: CALL INTERP ($1, $2, X, Y, A, F, I, J).

2.0 Category: FORTRAN subroutine, GADS accessory.

3.0 Purpose: To perform linear interpolation.

4.0 Variables:

4.1 Explicit Inputs:

$1	Error return. This return is invoked whenever $A<X(I)$.
$2	Error return. This return is invoked whenever $A>X(J)$.
X	Table of abscissae.
Y	Table of ordinates.
A	Argument abscissa.
I	Lower starting location for search.
J	Upper limit location for search. The search goes from X(I) to X(J), inclusive. Thus, $X(I) \leq A \leq X(J)$ for a successful return.

4.2 Explicit Outputs:

F	Interpolated function.

Module LOCALI. (Local to inertial transformation)

1.0 Calling Sequence: CALL LOCALI (BL,BI,RA,DECL).

2.0 Category: FORTRAN subroutine, GADS accessory.

3.0 Purpose: To transform from local north-east-vertical system to vernal equinox system.

4.0 Variables:

4.1 Explicit Inputs:

BL(3)	Cartesian vector in local coorindates.
RA	Radians longitude of sub-sattelite point.
DECL	Radians latitude of sub-sattelite point.

4.2 Explicit Outputs:

BI(3)	Cartesian vector in inertial coordinates.

4.3 Intermediate Variables:

ANGL1	First rotation, about +y axis.
CO1	Cosine ANGL1.
SI1	Sine ANGL1.
CO2	Cosine RA.
SI2	Sine RA.

5.0 Notes: The transformation from local north-east-down system to the GEI vernal equinox system achieved by two successive rotations. From Figure 14 it is eveident that the first rotation can be ANGL1 about the positive y axis. In terms of matrices,

$$\text{First rotation} = \begin{bmatrix} CO1 & 0 & SI1 \\ 0 & 1 & 0 \\ -SI1 & 0 & CO1 \end{bmatrix}$$

$$\text{Second rotation} = \begin{bmatrix} CO2 & -SI2 & 0 \\ SI2 & CO2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The product is:

$$\begin{bmatrix} CO2 \cdot CO1 & -SI2 & CO2 \cdot SI1 \\ SI2 \cdot CO1 & CO2 & SI2 \cdot SI1 \\ -SI1 & 0 & CO1 \end{bmatrix}$$

Module LOGCAL. (Boolean logical operations)

1.0 Calling Sequence: See Usage.

2.0 Category: SLEUTH II function, GADS support.

3.0 Purpose: To provide Boolean logic support.

4.0 Variables:

4.1 Input Variables:

I	First argument
J	Second argument

4.2 Output Variables:

This module returns a single variable according to the FORTRAN function rules.

5.0 Usage and Illustrative Examples:

Example 1. Logical "OR", fixed and floating point arithmetic, respectively:

```
K=KOR(IJ)
FK=OR(IJ)
```

Example 2. Logical "AND", fixed and floating point arithmetic respectively:

```
K=KAND(IJ)
FK=AND(IJ)
```

Example 3. Logical "exclusive OR", fixed and floating point arithmetic, respectively:

```
K=KXOR(IJ)
FK=XOR(IJ).
```

6.0 Other Entry Points: The following examples illustrate additional entry points:

Example 1. Load from I the partial word as per the J indicator:

L=LODPRT(IJ)
FL=PRTLOD(IJ)

The first example is used in fixed point arithmetic, the second, in floating. Refer to Figure 13 for J indicator codes. This figure is taken from Reference 8 which should be consulted for details concerning UNIVAC 1108 partial word capability.

Example 2. Load current address without sign extension:

L=LODPRT(I, 16_8)

Example 3. Load current address with sign extension:

L=LODPRT(I17_8)

Example 4. Store partial word from I to K as per J indicator:

CALL PRTSTR(I,K,J).

See Figure 13 and Reference 8 for a list of J operators.

Module MATM. (Matrix multiplication)

1.0 Calling Sequence: CALL MATM(A,B,C,N,L,M).

2.0 Category: FORTRAN subroutine, GADS support.

3.0 Purpose: Matrix multiplication.

4.0 Variables:

4.1 Explicit Inputs:

A(N,L)	First matrix.
B(L,M)	Second matrix.
N	Number of rows in A.
L	Number of columns in A, rows in B.
M	Number of columns in B.

4.2 Explicit Outputs:

C(N,M)	Matrix product A * B.

5.0 Notes: All matrices are handled so that the mathematical symbol B_{ij} corresponds to B(I, J).

Module MOVE. (Fast data transfer)

1.0 Calling Sequence: CALL MOVE(A, B, N, NA, NB).

2.0 Category: SLEUTH II subroutine, GADS support.

3.0 Purpose: to move data, especially when speed or convenience is needed.

4.0 Variables:

A	first word address of data source.
B	first word address of data destination.
N	number of data to be moved.
NA	difference between addresses of successive data source locations.
NB	difference between addresses of successive data destination locations.

5.0 Notes:

5.1 Efficiency: on the UNIVAC 1108, MOVE is more efficient than a "DO" loop when N < 5.

5.2 Features: NA and/or NB may be positive, negative, or zero.

5.3 Usage and Illustrative Exanples: the following 4 examples show a "DO" loop followed by an equivalent MOVE statement. The examples refer to variables given in the "DIMENSION" statement.

```
DIMENSION U(40), V(40), W(25,40), X(30,40), Y(20,10,40), Z(15,15,40)
```

Example 1:

```
DO 1 I = 1,40
V(I) = U(I)
```

```
CALL MOVE (U,V,40,1,1)
```

```
DO 2 I = 1,35,2
U(I+2) = 0.0
```

Example 2.

```
U(I+3) = 0.1

CALL MOVE (0.0, U(I+3), 18,0,2)
CALL MOVE (0.1, U(I+4), 18,0,2)

DO3 I = 1,12
X(f, I) = W(7, I)
```

Example 3.

```
X(9, I) = W(11, 41-I)

CALL MOVE (W(7, 1), X(5, 1), 12, 25, 30) (For
use of 25 and 30)
CALL MOVE (W(11, 40), X(9, 1), 12, -25, 30)
See the dimensions of W and X)

DO 4 I = 1, 9
Z(3, 4, I) = Y(5, I, 23)
```

Example 4.

```
U(I) = Y(6, 3, 41-I)

CALL MOVE (Y(5, 1, 23), Z(3, 4, 1), 9, 20, 225)
(225 = 15*15, see Z Dimension)
CALL MOVE (Y(6, 3, 40), U, 9, -200, 1)
(-200 = -20*10, see Y Dimension)
```

Module NDIST. (Normal distribution)

1.0 Calling Sequence: CALL NDIST(P, U).

2.0 Category: FORTRAN subroutine, GADS accessory.

3.0 Purpose: To compute normal distribution percentage points.

4.0 Variables:

4.1 Explicit Inputs:

P	The probability levels for which the percentage points are to be computed.

4.2 Explicit Outputs:

U	The normal distribution percentage point.

Module PAPER. (Paper display)

1.0 Calling Sequence: CALL PAPER.

2.0 Category: FORTRAN subroutine, GADS support.

3.0 Purpose: To generate a page (frame) of printed display in a high-speed printer.

4.0 Variables:

4.1 Explicit Inputs: None

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

BLANKS	Hollerith blanks.
BLANK1	Hollerith blanks except for left-most character. This character is a hollerith 1 and its purpose is to eject paper.
DASHES	Hollerith dashes.
DASH1	Hollerith dashes with exception similar to BLANK1.
DX	Input. Horizontal tic mark spacer. DX=0 means no tic marks.
DY	Input. Vertical tic mark spacer. DY = 0 means no tic marks.
IXRWRD	Word containing desired horizontal raster position.
IXR6TH	Horizontal raster position within word. Each word has 6 raster positions.
JOPERS	J operators needed to perform partial word handling. See Figure 13 and module LOGCAL.

JOPX	J operator for storing desired symbol in print array XGRID.
PRARR	Output. Print array area.
PRARR1	Same as PRARR.
VERTIC	Input. Vertically oriented tics for x-grid.
XFACTR	Input. Abscissa scale factor for conversion to raster counts.
XGRID	Output. Print area for horizontal grid.
XGRID1	See XGRID
XL	Input. Left limit of display area.
XR	Input. Bottom limit of display area.
YB	Input. Bottom limit of display area.
YT	Input. Top limit of display area.
YFACTR	Input. Ordinate scale factor for conversion to raster counts.
YGRID	Output. Vertical grid print area.
YGRID1	See YGRID.

Module POINTV. (Plot a point)

1.0 Calling Sequence: CALL POINTV(X, Y, NSYMB).

2.0 Category: FORTRAN subroutine, GADS support.

3.0 Purpose: To replace GSFC's POINTV in order to display on a high-speed printer.

4.0 Variables:

4.1 Explicit Inputs:

X	Abscissa of desired display.
Y	Ordinate of desired display.
NSYMB	Symbol selector. See Table 4.

4.2 Explicit Outputs: None.

4.3 Intermediate and Implicit I/O Variables:

BLANKS	Hollerith blanks.
IXR	Abscissa raster count.
IXRWRD	Word containing desired horizontal raster position.
IXR6TH	Horizontal raster position within word. Each word has 6 raster positions.
JOPERS	J operators needed to perform partial word handling. See Figure 13 and module LOGCAL.
JOPX	J operator for storing desired symbol in print array PRARR.
JOPS	J operator for extracting desired symbol from symbol library SYMBLS.

PRARR	Output. Print array area.
PRARR1	Same as PRARR.
QUESTN	Special symbol used in place of desired SYMBOL whenever the computed position (site) is occupied.
RESSYM	Resident symbol occupying computed position.
SYMBLS	Input. Library of packed hollerith symbols for plotting on high-speed printer. See module GADSDV and Table 4. This set of symbols conforms as much as possible to the one used by GSFC's POINTV as shown on page II-81, Reference 10.
XFACTR	Input. Abscissa scale factor for conversion to horizontal raster counts.
XL	Input. Left limit of display area.
XR	Input. Right limit of display area.
YB	Input. Bottom limit of display area.
YT	Input. Top limit of display area.
YFACTR	Input. Ordinate scale factor for conversion to vertical raster counts.

Module PWHAND. (Partial word handling)

1.0 Calling Sequence: See Section 2, Usage.

2.0 Category: SLEUTH II subroutine, GADS support.

3.0 Purpose: To pack and unpack arrays.

4.0 Variables:

S	Source address.
D	Destination address.
N	Item number or pointer index in array.
M	Half, third, or sixth-word selector: M=2 - half word, M=3 - third word, M=6 - sixth word. Note that the variable M need not be considered when using the various alternate entry points. See 5.2, Usage.

5.0 Notes:

5.1 Need for PWHAND:

The executive activities in GADS depend on numerous switch linkages or chains. These chains are the genetic matter of the GADS executive decision process. Because of their large number, it is clearly desirable to maintain them in packed form whenever possible.

5.2 Usage:

Assume that it is necessary to pack the array LINKF into array NCF by third words. See Figure 12. Hence, LINKF and NCF are the source and destinations, respectively. To pack the first 7 words the following sequence would be used:

```
             DO 10 I = 1,7
10           CALL PWDS(LINKF,NCF,I,3).
```

Conversely, to load LINKF from NCF, the sequence would be:

```
             DO 10 = 1,7
10           CALL PWDL(NCF,LINKF,I,3).
```

Note, that PWDS stores (packs) while PWDL loads (unpacks). More efficient use of this module is afforded by the multiple entry points. That is:

```
CALL PWDS(S,D,N,2)
CALL PWDS(S,D,N,3)
CALL PWDS(S,D,N,6)
```

are equivalent to:

```
CALL SHALVS(S,D,N,X)
CALL STHRDS(S,D,N,X)
CALL SSXTHS(S,D,N,X)
```

respectively. The variable X will be ignored and this type of call will result in a slightly reduced expenditure of time.

Similarly, to load halves, thirds, and sixths:

```
CALL LHALVS(S,D,N,X)
CALL LTHRDS(S,D,N,X)
CALL LSXTHS(S,D,N,C)
```

respectively.

Module RADECL. (Right ascension, declination)

1.0 Calling Sequence: CALL RADECL (X, RA, DEC).

2.0 Category: FORTRAN subroutine, GADS accessory.

3.0 Purpose: To compute right ascension and declination, given the unit vector in the vernal equinox system.

4.0 Variables:

4.1 Explicit Inputs:

X(3)	Cartesian unit vector.

4.2 Explicit Outputs:

RA	Right ascension, radians in the range $(0, 2\pi)$. See 5.0, Notes.
DEC	Declination, radians in the range $(-\pi, +\pi)$.

4.3 Intermediate Variables:

Temp 1	Magnitude of X projected onto equatorial or x-y plane.

5.0 Notes: When the projection of X onto the x-y plane is very small, i.e., less than 1.0×10^{-9}, the declination is set to $+\pi/2$ or $-\pi/2$, depending on whether X(3) is positive or negative. In this case the right ascension is not properly defined and is arbitrarily set to $+\pi$ or $-\pi$ depending on whether X(2) is positive or negative.

Moreover, whenever $|X(1)|$ is less than 1.0×10^{-9}, the right ascension is set to $+\pi/2$ or $-\pi/2$ depending on whether X(2) is positive or negative.

A discontinuity in RA will be observed if the vector X passes nearly or exactly over the north (or south).

Module RANDOM. (Uniform pseudo-random number generator)

1.0 Calling Sequence: CALL RANDOM(X).

2.0 Category: SLEUTH II subroutine, GADS accessory.

3.0 Purpose: To generate random numbers with approximately uniform distribution.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs:

X   Pseudo-random number in the range (0-1) with approximately uniform distribution.

5.0 Method: The multiplicative congruential method in References 11 and 12, is

$$K_i = aK_{i-1} + c \pmod{M};\ i = 1, 2, 3 \ldots$$

In the present module,

$K_0 = a,$

$a = 343277244615_8,$

$a = 30517578125_{10},$

$c = 0,$

$M = 2^{35},$

$0 < K_i < 2^{35}.$

The sequence K_1, K_2, ... is in the open interval $(0, 2^{35})$ and has a cycle of 2^{35} or less. In order to obtain floating point output, X_i, the 27 most significant bits of K_i are retained and combined with an exponent such that

$$0.0 < X_i < 1.0$$

It is explained in Reference 21 that if the cycle is to be 2^{35}, must be equal to $2^{35}-31$, the largest prime less than 2^{35}. Consequently, the cycle of this algorithm is less than 2^{35}.

Module SEARCH.

1.0 Calling Sequence: CALL SEARCH (K, L, M, N, $).

2.0 Category: FORTRAN subroutine, GADS support.

3.0 Purpose: To search K words in an array M to identify the argument L.

4.0 Variables:

4.1 Explicit Inputs:

K	Number of consecutive cells to investigate.
L	The argument to match bit-by-bit.
M	Array to search.
$	Return address if search is unsuccessful.

4.2 Explicit Outputs:

N	The pointer index if a match is successful.

Module SHADOW.

1.0 Calling Sequence: CALL SHADOW ($).

2.0 Category: FORTRAN subroutine, GADS accessory.

3.0 Purpose: To determine the occultation status (shade) of a specific sensor.

4.0 Variables:

4.1 Explicit Inputs: None.

4.2 Explicit Outputs:

$	Alternate return if the sensor is not occulted.

4.3 Intermediate and Implicit I/O Variables:

AVERTX(2,4)	Input. Spherical coordinates for the vertices: Φ_j = AVERTX(1,j) Θ_j = AVERTX(2,j).
CSHADA	Critical shadow angle, degrees. See Note 4.
COSA	Cos (A), where A is one of the three angles A, B, C discussed in 5.0, Notes.
FTEMP	Input. The theoretical output of a cosine sensor operating on SPRIME.
II	The total number of triangles.
INTEP2	Input. See module GADS/EPED.
J	Pointer index for vertices.
NCALLS	Input and output. Call counter.
RPD	Input. Radians per degree.

SCROSV(3)	The sines of the three great arcs joining the source and the three vertices of a given triangle.
SDOTV(3)	The cosines of the same arcs. See the preceding line.
SHADAN	Shadow angle. The accumulator for A+B+C. (See Notes)
SHADCR(4)	Input shadow criteria: SHADOW(1) = criterion for testing SHADAN, SHADOW(2) = criterion for testing FTEMP (this is a coarse occultation criterion), SHADAN(3) = not used, SHADAN(4) - not used.
SPRIME(3)	Input. The sensor operand (see module GADF01).
SOURCE(3)	Input. See EQUIVALENCE statement. The user specifies his source of radiation by means of the said statement.
VCONE	Coarse criterion for FTEMP.
VDOTV(3, 2)	Cosines of the great arcs defining the spherical triangles mesh: VDOTV(j, i); j = 1, 2, 3 -- three cosines for triangle i.
VERTEX(3, 4)	Cartesian components of the vertex vectors: VERTEX(k, j); k = 1, 2, 3 -- three components for vertex j.
ALL OTHERS	For all other variables, refer to module GADF01.

5.0 Notes:

As illustrated in Figure 19, the center of coordinates is the sensor and is assumed to be infinitesimal in size. The object causing the occultation is defined on the unit sphere by the vertices V_j which are finite in number and are joined by great arcs. (The user provides these vertices as discussed below.) This object is divided into a set of adjacent non-overlapping spherical triangles.

Whether the sensor (origin) is in view of the source S depends on whether the vector S is inside any of the triangles. The approach taken in this module is to compute the angles A, B, and C shown in Figure 19b or 19c. When S is inside a given triangle, as in Figure 19c, $A+B+C = 360°$; when S is outside, as in Figure 19b, A+B+C is less than 360°. Thus the method consists of comparing A+B+C to 360°. The present analysis is not dependent on the sensor being physically at the origin of the coordinates.

5.1 Preliminary Calculations:

To perform the preceding analysis it is necessary to determine the vectors V defining the triangle mesh and the cosines of the great arcs α, β, γ. Since these quantities are constants in the spacecraft coordinate system, they need only be computed once (in the first call).

Number the triangles with the symbol i; $i = 1, 2, \ldots II$, the vertices with $j = 1, 2, \ldots II+2$. (This can always be done.) Then V_{jk} is the k component of the j vertex. In this module, the components of j vertex for example, are computed from the angles Φ_j and Θ_j which the user must supply (in degrees). These angles are defined in the same way that sensor orientation angles are defined. See Figure 16.

The components are given by:

$$V_{j1} = \cos(\Theta_j) \cdot \cos(\Phi_j) \tag{1a}$$

$$V_{j2} = \cos(\Theta_j) \cdot \sin(\Phi_j) \tag{1b}$$

$$V_{j3} = \sin(\Theta_j). \tag{1c}$$

Then the cosines of the great arcs are:

$$\cos(\alpha_i) = V_j \cdot V_{j+1};\ j = i \tag{2a}$$

$$\cos(\beta_i) = V_{j+1} \cdot V_{j+2};\ j = i \tag{2b}$$

$$\cos(\gamma_i) = V_{j+2} \cdot V_j;\ j = i \tag{2c}$$

5.2 Determination of A+B+C:

The angles A, B, and C are computed using the spherical law of cosines:

$$\cos a = \cos b \cdot \cos c + \sin b \cdot \sin c \cdot \cos A \tag{3}$$

or

$$\cos A = (\cos a - \cos b \cdot \cos c)/\sin b \cdot \sin c. \tag{4}$$

Expressions for B and C may be obtained by cyclic permutation of a, b, and c. To apply equation (4), the cosines and sines of the great arcs joining S to the three vertices of each triangle are required. Consider, for example, triangle (V1, V2, V3), namely triangle 1:

$$\cos a = \alpha_1 \tag{5a}$$

$$\cos b = S \cdot V_1 \tag{5b}$$

$$\cos c = S \cdot V_2. \tag{5c}$$

The sines can then be obtained from:

$$\sin x = (1 - \cos^2 x)^{\frac{1}{2}}. \tag{6}$$

Since this module is not invoked whenever S is more than 90^o from any vertex, there is no danger of losing the correct quadrant in formula (6). These sines are stored in the array SCROSV.

For any given triangle it is necessary to compute all three angles A, B, and C subtended at S by each of its sides α, β, and γ, with the sum A+B+C the required quantity.

5.3 Usage:

In order for this module to function properly, the user should perform the following items:

a. Determine the shape of the object causing geometric shadowing in the body-fixed frame of reference. Consider only its projection on the unit sphere centered at the sensor. See Figure 19.

b. Choose the minimum number of vertices which, when joined by great arcs, best describe the said projection.

c. Add great arcs in the interior of the figure thus formed so that it appears formed by non-overlapping adjacent triangles.

d. Number the triangles sequentially beginning with 1 and ending with II.

e. Number the vertices beginning with 1 and ending with II+2 in such a way that vertices V_j, V_{j+1}, and V_{j+2} belong to the triangle.

f. Insure that the angles Φ_j and Θ_j, determined independently, are available upon entry to this program. These should be in degrees and stored so that:

$$\Phi_j = \text{AVERTX}(1,j)$$

$$\Theta_j = \text{AVERTX}(2,j).$$

Upon entry to this module, the variable SOURCE should contain the cartesian components (with respect to the body-fixed system of coordinates) of the normalized vector pointing toward the source of radiation. This is accomplished by the EQUIVALENCE (SOURCE, SPRIME) statement. The variable SPRIME is discussed in module GADF01.

Note that the user may relax the criterion 360° ; because of rounding errors it is preferable to set the criterion to 359°. Hence the variable SHADCR(1). Furthermore, the variable SHADCR(2) provides the user with a gross criterion for occultation. That is, whenever the predicted function, assumed to be that of a cosine sensor, is less than VCONE the remainder of the logic is unnecessary. In such a case, the angle between SPRIME and the sensor exceeds SHADCR(2).

Module SIMP2. (Simpson Integration)

1.0 Calling Sequence: See Usage.

2.0 Category: FORTRAN function, GADS accessory.

3.0 Purpose: To evaluate $\int_{x_1}^{x_{n+1}} F(x)dx$ where F(x) is given in tabular form at N+1 equally spaced points.

4.0 Variables:

4.1 Input Variables:

F	Array of N+1 function values F(x).
A	Lower limit of integration, x_1
DELX	Length of integration step.
N	Number of intervals over which the function is to be integrated.

4.2 Output:

The function SIMP2 returns a single result according to the rules governing FORTRAN functions.

5.0 Notes:

5.1 Method:

Normally Simpson's rule is used to integrate over N = 2k intervals (i.e., N+1 points).

SIMP2 uses Simpson's rule with the modification that integration over an odd number of intervals is possible. The number of intervals, N, is tested to determine whether it is even or odd. If N is even, integration proceeds as in SIMP1. If N is odd, a second degree polynomial is fit through the last three points in the range of integration, X_{2k}, X_{2k+1}, X_{2k+2}. This polynomial is then integrated from X_{2k+1} to X_{2k+2}, i.e., over the last interval. The result is added to the result of the conventional Simpson integration over the first 2k intervals.

5.2 Usage:

SIMP2 (F, A, DELX, N) is used in a floating point arithmetic statement.

5.3 Accuracy:

The inherent error using Simpson's rule when N is even is given by

$$\frac{X_{n+1} - X_1}{180} (\Delta x)^4 F^{(4)} (\alpha), \; X_1 < \alpha < X_{n+1}$$

When N is odd, the inherent error is still of order $(\Delta x)^4$.

5.4 Restrictions:

SIMP2 is set to zero is $N < 2$.

Module SORBIT. (Simulated Orbit)

1.0 Calling Sequence: CALL SORBIT(TIME, TRUEA, RAD, XW, X, OMEGA, FINCL, TRUEAR, ROOT).

2.0 Category: FORTRAN subroutine, GADS accessory.

3.0 Purpose: To generate artificial orbital ephemeris and some related data.

4.0 Variables:

4.1 Explicit Inputs:

TIME	Ignored.
TRUEA	True anomaly, radians.
RAD	Orbital radius vector, kilometers.
OMEGA	Right ascension of orbital mode, radians.
FINCL	Orbital inclination, radians.
TRUEAR	True anomaly rate, radians per sec.

4.2 Explicit Outputs:

XW	Radius vector in orbit plane, kilometers.
X	Radius vector in vernal equinox coordinate system, kilometers.
RDOT	Velocity vector in vernal equinox system, kilometers per second.

5.0 Explanation: This module is used to generate reasonable arguments for the FIELDG modules. Hence all orbits are assumed circular.

Module VCROS. (Vector cross product)

1.0 Calling Sequence: CALL VCROS (A, B, C, $1, SIN, K).

2.0 Category: FORTRAN subroutine, GADS accessory.

3.0 Purpose: To compute the vector cross product and, if desired, the sine of the included angle.

4.0 Variables:

4.1 Explicit Inputs:

A	First cartesian vector
B	Second cartesian vector
$1	Error return. This return is invoked whenever the cross product has a vanishing magnitude.
K	Normalization request: K = 0 - no request K $\neq$ 0 - normalize the result C and compute the sine of the included angle.

4.2 Explicit Outputs:

C	Cross product A × B.
SIN	Sine of included angle (A, B), computed only if K $\neq$ 0.

Module TDIST. (Student "t" Distribution)

1.0 Calling Sequence: CALL TDIST(ALPHA, NDEGF, T).

2.0 Category: FORTRAN subroutine, GADS accessory.

3.0 Purpose: To compute t-distribution percentage points; i.e., to compute the value of t_α necessary for determining interval estimates of the parameters.

4.0 Variables:

4.1 Explicit Inputs:

ALPHA	The probability level for which the percentage points are to be computed.
NDEGF	The number of degrees of freedom; i.e., the number of data points minus the number of parameters.

4.2 Explicit Outputs:

T	The value of t corresponding to ALPHA and NDEGF, i.e., t_α.